Secure Roaming and Handover Procedures in Wireless Access Networks

Vom Fachgebiet Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

von
Ulrike Meyer
aus Leutkirch

Referenten: Prof. Dr. Johannes Buchmann
Prof. Dr. Susanne Wetzel

Tag der Einreichung: 30. August 2005
Tag der mündlichen Prüfung: 20. Dezember 2005

Darmstadt, 2005
Hochschulkennziffer: D17
To my godmother Edeltrud
Abstract

A growing number of wireless technologies and providers, as well as users’ increasing need and desire to be connected and reachable at all times, call for solutions that enable inter-operation between providers and technologies. Roaming procedures enable wireless access in areas that are covered by network providers with which the user does not have any prior arrangements. Handover procedures enable the maintenance of ongoing connections while a user moves across different wireless access networks.

The goal of this thesis is to model the security challenges imposed on infrastructure-based wireless access networks by inter-provider and inter-system roaming and handover procedures, to analyze current solutions in this model, and to develop new security solutions.

In the first two parts of this thesis, the theoretical parts, we present new models for handover and roaming as well as new security solutions in a technology-independent way:

In part II, the main part of this thesis, we develop a formal model for security-context transfer on various types of inter-provider handover procedures. As opposed to previous work [186, 177, 74, 185, 75, 162] our model explicitly captures security-context transfers on subsequent handover and handover after roaming. We present a thorough threat analysis of security solutions that are based on security-context transfers during inter-provider handover and define new security requirements based on this analysis. As state-of-the-art handover procedures do not meet our requirements, we present a new history-enriched, policy-based approach to enhance security-context transfers on inter-provider and inter-system handover. The main advantage of our new approach is that it allows mobile devices and networks to express policies with respect to whether or not a handover should take place, dependent on the history of previously used security mechanisms that is included in the security-context transfer. This protects users and providers from the impact of previously used weak security mechanisms. Moreover, in our procedures, users and networks can enforce policies with respect to the security mechanisms used after the current handover. This protects the handover participants from the impact of future use of weak security mechanisms.

Furthermore, in part I, we model, classify and discuss roaming authentication protocols for wireless access networks in a technology-independent way. We introduce a new public-key-based approach for authentication upon roaming. As opposed to previous work on inter-provider roaming (e.g., [62, 9, 156]), our approach does not require a secure channel between the home and the foreign network. Moreover, as opposed to other public-key-based solutions [81, 29], in our approach a mobile device is not required to obtain and validate a
Abstract

In the third part of this thesis, we analyze the roaming and handover procedures within and between GSM and UMTS. In particular, we present a man-in-the-middle attack against the authentication and key agreement based on UMTS-authentication vectors. This attack is enabled by a vulnerability in the inter-operation of UMTS with GSM. Furthermore, we discuss whether the inter-system handover procedures between GSM and UMTS meet the security requirements newly defined in the theoretical part. We show that attacks against the GSM encryption and a man-in-the-middle attack against the GSM authentication and key agreement have an impact on the security of a connection between a user and a UMTS network if a user is handed back and forth between UMTS and GSM.

Finally, in the fourth part, we apply our new security solutions to roaming and handover between IEEE 802.11 WLANs. In particular, we present a roaming authentication protocol EAP-TLS-KS that implements the new roaming solution. Furthermore we detail how the history-enriched, policy-based approach for inter-provider handover can be implemented in the WLAN context.
Zusammenfassung


Das Ziel der vorliegenden Arbeit ist es, die Sicherheitsprobleme, die durch Handover- und Roamingprozeduren zwischen verschiedenen Betreibern und Technologien für infrastrukturbasierte drahtlose Netze entstehen, zu modellieren, bestehende Lösungen in diesem Modell zu analysieren und neue Sicherheitslösungen zu entwickeln.

In den ersten beiden Teilen dieser Arbeit, den theoretischen Teilen, werden neue technologieunabhängige Modelle für Roaming und Handover eingeführt.

die auf der Benutzung schwacher Sicherheitsmechanismen nach einem Handover beruhen.

Darüber hinaus werden in Teil I der Arbeit Authentisierungs- und Schlüsselvereinbarungsprotokolle für Roaming modelliert, klassifiziert und diskutiert und ein neuer Ansatz für public-key-basiertes Roaming zwischen verschiedenen Netzbetreibern entwickelt. Im Gegensatz zu anderen Arbeiten auf diesem Gebiet (z.B. [62, 9, 156]), benötigt unser Ansatz keinen sicheren Kanal zwischen dem Heimnetz und einem Fremdnetz. Ausserdem muss in unserem Ansatz ein mobiles Endgerät keinerlei Ketten von Zertifikaten konstruieren und auswerten, was ein Vorteil gegenüber anderen public-key-basierten Ansätzen (z.B. [81, 29]) ist.


Schließlich werden im vierten Teil der Arbeit die neuen Roaming- und Handoverlösungen auf drahtlose lokale Netze nach IEEE 802.11 angewandt. Wir stellen ein neues Roamingprotokoll EAP-TLS-KS vor und beschreiben, wie der neue geschichts- und policy-basierte Sicherheitskontexttransfer im Fall drahtloser lokaler Netze implementiert werden kann.
Acknowledgments

Many people have contributed directly or indirectly to the completion of this work. In particular, I want to thank

Prof. Dr. Johannes Buchmann for his advice and support, for encouraging me and believing in my abilities, and for his exceptional talent to amplify all my moods.

Prof. Dr. Susanne Wetzel for creating a wonderfully productive working atmosphere, investing her time in many discussions that not only clarified my thoughts and added new ideas to every part of this thesis but also were a pleasure in themselves, for her admirable sense for perfection, and for supporting and advising me in every thinkable way.

My colleagues at Technische Universität Darmstadt, as well as the faculty and students of the GK “Systemintegration für Ubiquitäres Rechnen in der Informationstechnik” and, in particular, Kira Kastell for interesting discussions and working with me during the first phase of this thesis.

The faculty and students of the LSS Lab for lively discussions on my thesis topic and enriching my daily life at the Stevens Institute. In particular, Jared Cordasco for patiently sharing his Linux and networking knowledge with me and making our joint work on the content of Chapter 11 a pleasure.

Birgit Henhapl and Markus Maurer for jointly fighting their way through this thesis and sacrificing their weekends to contribute to the final version with various valuable comments. Werner Backes for his very helpful comments on several versions of this thesis and for contributing to my mental stability. Tara Fuyter for correcting my English and trying to teach me the correct hyphenation of compound words.

Jan Rübel for his endless patience and emotional support throughout the last turbulent years, for taking care of me, not neglecting my social life, for ensuring excellent food supply, and for last minute emergency proofreading and formatting.

My parents for their continuous, unintrusive, and selfless support. My sister and all my friends for putting up with me during this eventful time, re-charging my power, and dragging me away from my work. I would never have made it without your support!

Thank you!

I also want to thank the German Research Foundation (DFG), the German Academic Exchange Service (DAAD) and the Wireless Network Security Center (WinSec) for financially supporting my work.
Erklärung


**Wissenschaftlicher Werdegang des Verfassers in Kurzfassung**

<table>
<thead>
<tr>
<th>Datum</th>
<th>Ereignis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oktober 1994 - Januar 2001</td>
<td>Studium der Mathematik mit Nebenfach Volkswirtschaftslehre, Universität Heidelberg</td>
</tr>
<tr>
<td>Januar 2001</td>
<td>Diplom, Diplomarbeit: <em>Normenkörper</em>, Universität Heidelberg</td>
</tr>
<tr>
<td>Dezember 2001 - Dezember 2005</td>
<td>Doktorandin am Lehrstuhl für Kryptographie und Computeralgebra, Prof. J. Buchmann, Fachbereich Informatik, Technische Universität Darmstadt</td>
</tr>
<tr>
<td>Januar 2001 - Dezember 2004</td>
<td>Stipendiatin im Graduiertenkolleg <em>Systemintegration für Ubiquitäres Rechnen in der Informatik</em> der DFG</td>
</tr>
<tr>
<td>März 2004 - November 2005</td>
<td>Mehrere Forschungsaufenthalte bei Prof. S. Wetzel, am Stevens Institute of Technology, Hoboken, NJ, USA, mit finanzieller Unterstützung der DFG im Rahmen des Graduiertenkollegsstipendiums, des DAAD durch ein Kurzzeitstipendium für Doktoranden und des Wireless Network Security Centers (WinSec)</td>
</tr>
</tbody>
</table>
# Contents

Introduction 1

I Wireless Access Networks and Roaming 9
1 System Model and Security Model 11
   1.1 System Model 12
   1.2 Security Model 15
   1.3 Security-Mechanism Negotiation 28

2 Roaming 39
   2.1 Inter-Provider Roaming - Modeling and Classification 41
   2.2 Enhancing Roaming Protocols by Means of Secret-Sharing 50
   2.3 Roaming Across Different Access Technologies 52
   2.4 Related Issues and Future Directions of Research 54
   2.5 Related Work 57
   2.6 Conclusion 60

II Handover 63
3 Handover—Model Procedures and Security Solutions 65
   3.1 Model Procedures 68
   3.2 The Security Challenge and Solutions 78
   3.3 Related Work 98
   3.4 Conclusion 99

4 Threat Analysis for Security-Context Transfer 101
   4.1 Attack Trees 103
   4.2 First-Order—HN as Anchor—SCT with Key Derivation 104
   4.3 k-th-order Handover—SCT with Key Derivation 125
   4.4 Differences on SCT with Key Agreement 140
   4.5 Related Work 141
4.6 Conclusion .................................................. 142

5 History-Enriched Policy-Based SCT ........................................... 143
  5.1 HEPB SCT with Key Derivation ......................................... 144
  5.2 HEPB SCT with Key Agreement ......................................... 164
  5.3 Using Secret Sharing for Key Agreement ............................. 165
  5.4 Conclusion ..................................................... 167

6 Extensions and Related Issues ............................................... 169
  6.1 Inter-System Handover ............................................... 170
  6.2 Related Work on Inter-System Handover ............................. 176
  6.3 Intra-Provider, Intra-System Handover ......................... 178
  6.4 Mobility Prediction and Handover ................................. 179
  6.5 Location Management ........................................... 179
  6.6 Conclusion ..................................................... 180

III Handover and Roaming within and between GSM and UMTS ....... 181

7 Inter-provider Roaming within GSM and UMTS ......................... 183
  7.1 GSM Inter-Provider Roaming ....................................... 184
  7.2 GSM Intra-Provider Roaming ....................................... 187
  7.3 UMTS Inter-Provider Roaming ...................................... 189
  7.4 UMTS Intra-Provider Roaming ...................................... 192

8 Roaming Between GSM and UMTS .......................................... 195
  8.1 GSM/UMTS Inter-System Inter-Provider Roaming ................ 196
  8.2 Intra-Provider Roaming within a Mixed-Mode Network .......... 200
  8.3 Man-in-the-Middle Attack on UMTS ................................ 201
  8.4 Conclusion ..................................................... 207

9 Handover within and between GSM and UMTS ............................ 209
  9.1 Analysis of Intra-Provider Handover in GSM .................... 210
  9.2 Intra-Provider Handover within UTRAN ............................ 213
  9.3 Inter-System Handover between GSM and UMTS .................... 214
  9.4 Impact of GSM Vulnerabilities on UMTS via Handover .......... 219
  9.5 Conclusion ..................................................... 225

IV Inter-Provider Handover and Roaming in WLANs ....................... 227

10 System Model and WLAN Security ......................................... 229
  10.1 System Model .................................................. 230
  10.2 WEP ............................................................ 230
## Contents

10.3 Overview on 802.11i ........................................ 231
10.4 Conclusion ............................................. 235

### 11 The New Protocol EAP-TLS-KS
11.1 Overview of EAP-TLS ..................................... 238
11.2 EAP-TLS with Key Splitting ................................. 241
11.3 Related Work ........................................... 254
11.4 Conclusion ............................................. 257

### 12 History-Enriched Policy-Based SCT for WLAN
12.1 HEPB Handover in the WLAN Context ..................... 260
12.2 The Candidate Access Router Discovery (CARD) ............. 261
12.3 The Context Transfer Protocol (CXTP) ....................... 261
12.4 Implementing HEPB Handover Using CARD and CXTP ....... 263
12.5 Conclusion ............................................. 266

Conclusion .................................................. 269

### A Attack Trees

275
List of Figures

1 Notations Used in Figures .................................................. xxi
1.1 GSM Network with a Hierarchical Backbone .......................... 13
1.2 WLAN Network with a Flat Backbone ................................. 14
1.3 System Model of a Wireless Access Network ....................... 14
1.4 Wireless Access Network Components in the ISO/OSI Model ........ 15
1.5 Entities and Relationships .............................................. 16
1.6 EIPE and NAP Coincide ................................................... 25
1.7 EIPE and NAP are Different ............................................. 25
1.8 EIPE = NAP and ISO/OSI ............................................... 25
1.9 EIPE ≠ NAP and ISO/OSI ................................................. 25
1.10 Connection Establishment (See Page 1 for Notations Used in Figures) ... 28
1.11 Message Exchange between A and B upon Method 5 ................ 35
1.12 A’s Decision in Step i of Method 5 ($P^i_A$ in Figure 1.11) ........ 36
2.1 Type1 Roaming Procedures (See page xxi for Notations Used in Figures) . 49
2.2 Type2 Roaming Procedure .................................................. 49
2.3 Type3 Roaming Procedures ................................................ 51
3.1 Order and Anchor Type of a Handover .................................. 66
3.2 Inter-Provider Handover Scenario ...................................... 70
3.3 First-Order Network-Initiated Handover Procedure with HN as Anchor .. 71
3.4 HN-Controlled Subsequent Handover .................................. 73
3.5 General Network-Initiated Handover Procedure ..................... 76
3.6 Mobile-Initiated, HN as Anchor, First-Order, HN Notified by MD ....... 77
3.7 Mobile-Initiated, HN as Anchor, First-Order, HN Notified by DEST ... 78
3.8 Mobile-Initiated $k$-th-order Handover, HCN Notified by DEST$_k$ ........ 79
3.9 Mobile-Initiated $k$-th-order Handover, HCN Notified by MD .......... 80
3.10 Full Authentication via DEST$_k$ in the Network-Initiated Case .......... 81
3.11 Pre-Authentication via NAP$_{SRC_k}$ in the Network-Initiated Case .......... 84
3.12 First-Order Network-Initiated Handover with HN as Anchor Network ... 87
3.13 Network-Initiated Handover in the General Case ...................... 89
3.14 SCT on Mobile-Initiated $k$-th-order Handover, HCN Notified by DEST$_k$ ... 93
3.15 SCT on Mobile-Initiated k-th-order Handover, HCN Notified by MD . . . . . 94
3.16 Network-Initiated Handover with SCT with Key Agreement and Authentica-
tion During Ongoing Connection ........................................... 96
3.17 Multiple Initial Key Generation for SCT with Key Agreement ............ 97
4.1 Root Attack Scenario ...................................................... 104
4.2 Regular Subgoal ......................................................... 104
4.3 Reappearing Subgoal .................................................... 104
4.4 B OR C ................................................................. 104
4.5 B AND C .............................................................. 104
4.6 (B AND C) OR D ...................................................... 104
4.7 Attack Tree for Root Attack Scenario RAS-1 ............................ 108
4.8 Attack Tree for Subgoal “Recover EK₀ without Knowledge of K₀,” ........ 109
4.9 Attack Tree for Subgoal “Reconstruct ke₀.” ............................ 110
4.10 Attack Tree for the Subgoal “Recover K₀ without Knowledge of EK₀, IK₀,
and K₁.” ................................................................. 111
4.11 Attack Tree for the Subgoal “Recover K₁ without Knowledge of EK₁, IK₁,
and K₀.” ................................................................. 111
4.12 Attack Tree for Subgoal “Totally Break em₁ without EK₀ and without Dis-
abling em₀.” ................................................................. 112
4.13 Attack Tree for Subgoal “Totally Break im₁ without EK₀ and without Dis-
abling em₀.” ................................................................. 113
4.14 Summary of BAMs and AMs for First-Order Handover with HN as Anchor 116
4.15 Summary of BAMs and AMs on k-th-order Handover ..................... 129
5.1 Details of “select DEST” of Figure 3.12 .................................. 147
5.2 Details of “DEST’s decision” of Figure 3.12 ............................... 148
6.1 Vertical Handover ......................................................... 170
6.2 Horizontal Handover ..................................................... 170
7.1 System Model and Storage of Security Information ......................... 184
7.2 GSM Authentication, Key Agreement, and Security-Mechanism Negotiation 188
7.3 UMTS System Model and Security Mechanism Endpoints .................. 189
7.4 UMTS Authentication, Key Agreement, and Security-Mechanism Negotiation 193
8.1 The Six Roaming Cases .................................................... 197
8.2 A SIM-Equipped MD Roams to UMTS (Case 3) ........................... 198
8.3 USIM-Equipped MD Roams to GSM (Case 4) .............................. 199
8.4 USIM-Equipped MD Roams to a Mixed-Mode Network (Case 5) .......... 200
8.5 GSM Subscriber Roaming to a Mixed-Mode Network (Case 5) ............ 200
8.6 Intra-Provider Roaming in a Mixed-Mode Network ....................... 201
8.7 Phase 1: Attacker Obtains Currently Valid AUTN ........................ 204
8.8 Phase 2: Attacker Impersonates Valid GSM Base Station to the Victim 205
List of Figures

9.1 USIM Handover Cases with Authenticating MSC = Anchor MSC 217
9.2 Additional USIM Handover Cases with Authenticating MSC ≠ Anchor MSC 217
9.3 SIM Handover Cases 218

10.1 System Model for an IEEE 802.11 WLAN 230
10.2 The Protocol Architecture in 802.11i 233
10.3 Overview of 802.11i 233
10.4 Key Hierarchy in 802.11i 234

11.1 Overview of the EAP-TLS Protocol 239
11.2 EAP-TLS with RSA 241
11.3 EAP-TLS with DHE 241
11.4 EAP-TLS-KS with RSA 246
11.5 EAP-TLS-KS with DHE-RSA 249
11.6 EAP-TLS-KS with DHE-DSS 250

12.1 Predictive CXTP 262
12.2 Reactive CXTP 262
12.3 Predictive HEBP-Based Procedure 264
12.4 Reactive HEBP-Based Procedure 265

A.1 Alternative Pairs of AND Nodes 275
A.2 Alternative Sets of AND Nodes That Make Use of a Common Node 275
A.3 Attack Tree for RAS-2 276
A.4 Subtree for Subgoal “Disable $e_{m_0}$”; For “Disable $i_{m_0}$” Replace $e_{m_0}$ with $i_{m_0}$ 276
A.5 Attack Tree for RAS-3 277
A.6 Subtree for Subgoal “Manipulate $e_{m_1}$”; For “Manipulate $i_{m_1}$” Replace $e_{m_1}$ with $i_{m_1}$ 278
A.7 Subtree for Subgoal Combination “$E_{K_0}$ without $K_1$ and $E_{K_0}$ without $E_{K_1}, I_{K_1}$”; For “$I_{K_0}$ without $K_1$ and $I_{K_0}$ without $E_{K_1}, I_{K_1}$” replace $E_{K_0}$ with $I_{K_0}$ 279
A.8 Subtree for Subgoal “Reconstruct $k_{e_1}$” 279
A.9 Subtree for Subgoal “Recover $K_1$ without $E_{K_1}$ and $I_{K_1}$”; For “Recover $K_1$ without $E_{K_1}$ and $I_{K_1}$” Replace $E_{K_1}$ with $I_{K_1}$ 280
A.10 Attack Tree for RAS-4 280
A.11 Subtree for “Impersonate NAP-DEST after a real handover command” 281
A.12 Subtree for Subgoal “Bids down $e_{m_1}$ to no encryption”; For Subtree “Bids down $i_{m_1}$ to no integrity protection” Replace $e_{m_1}$ with $i_{m_1}$ 282
A.13 Attack Tree for RAS-5 283
A.14 Attack Tree for RAS-6 284
A.15 Attack Tree for RAS-7 285
A.16 Attack Tree for RAS-8 286
A.17 Attack Tree for RAS-9 287
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.18</td>
<td>Subtree for “False handover detection”</td>
<td>288</td>
</tr>
<tr>
<td>A.19</td>
<td>Attack Tree for RAS-10</td>
<td>288</td>
</tr>
<tr>
<td>A.20</td>
<td>Attack Tree for RAS-11</td>
<td>288</td>
</tr>
<tr>
<td>A.21</td>
<td>BAM-1, BAM-2, BAM-3, BAM-7, BAM-8, BAM-9, and AM-1.</td>
<td>289</td>
</tr>
<tr>
<td>A.22</td>
<td>AM-2</td>
<td>289</td>
</tr>
<tr>
<td>A.23</td>
<td>AM-3</td>
<td>290</td>
</tr>
<tr>
<td>A.24</td>
<td>BAM-4, BAM-5, BAM-6, BAM-8, BAM-9, BAM-10, and AM-4.</td>
<td>291</td>
</tr>
<tr>
<td>A.25</td>
<td>AM-5</td>
<td>292</td>
</tr>
<tr>
<td>A.26</td>
<td>AM-6</td>
<td>293</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Types of Roaming Procedures</td>
<td>48</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview on Attacks and Attack Modules</td>
<td>121</td>
</tr>
<tr>
<td>4.2</td>
<td>Attack Modules and Requirements</td>
<td>124</td>
</tr>
<tr>
<td>4.3</td>
<td>Overview on Attacks and Attack Modules</td>
<td>135</td>
</tr>
<tr>
<td>4.4</td>
<td>Basic Attack Modules and Requirements</td>
<td>139</td>
</tr>
<tr>
<td>9.1</td>
<td>Candidates for New Attacks against GSM</td>
<td>212</td>
</tr>
<tr>
<td>9.2</td>
<td>Candidates for Attacks against UMTS</td>
<td>215</td>
</tr>
<tr>
<td>11.1</td>
<td>Comparison</td>
<td>251</td>
</tr>
</tbody>
</table>
## Glossary

### General Terms

- $||$ : Concatenation of bit strings
- $\oplus$ : Xor of bit strings
- $x \in_R X$ : $x$ randomly chosen from $X$
- $\varphi(\cdot)$ : Euler phi-function
- 3GPP : Third Generation Partnership Project
- $A$ : Technology-specific set of authentication protocols
- $a \in A$ : Authentication protocol
- AN : Anchor Network
- AS : Authentication Server
- CA : Certification Authority
- $CS$ : Technology-specific set of cipher suites
- $CS^* X|_{ssh_{k-1}}$ : Commitment to $CS X|_{ssh_{k-1}}$ by $X$
- $cs \in CS$ : Cipher suite
- $CS X|_{ssh_{k-1}}$ : Subset of $CS X$ allows after $k$-th order handover given $ssh_{k-1}$
- DEST : DESTination network
- EAP : Extensible Authentication Protocol
- EAP-TLS : EAP based on Transport Layer Security
- EAP-TLS-KS : EAP-TLS with Key Splitting
- EIPE : Encryption and Integrity-Protection Endpoint
- $EK$ : Encryption Key
- $EM$ : Technology-specific set of encryption mechanisms
- $em \in EM$ : Encryption mechanism
- FN : Foreign Network
- GSM : Global System for Mobile communications
- $h$ : Technology-specific maximal number of subsequent handover
HCN .................................................... Handover Controlling Network
HEPB ................................................... History-Enriched Policy-Based
history_{k−1} ........................................... Context history on a $k$-th order handover
HN ........................................................... Home Network
IK .......................................................... Integrity-Protection Key
IM ........................................................ Technology-specific set of integrity-protection mechanisms
im ∈ IM ................................................ Integrity-protection mechanism
ISDN ................................................ Integrated Services Digital Network
IMT-2000 ............................................ International Mobile Telecommunications 2000
\( K K_i \) .................................................. Master session keys
KA ..................................................... Technology-specific set of key agreement protocols
ka ∈ KA ............................................... Key-agreement protocol
KE ........................................................ Technology-specific set of key-establishment process
ke ∈ KE ............................................... Key-establishment process
KD ........................................................ Set of key-derivation functions
kd ........................................................ key-derivation function
NAP ........................................................ Network Access Point
MAC ................................................. Message Authentication Code
MDC .................................................. Modification Detection Code
MD ........................................................ Mobile Device
MiM ..................................................... Man-in-the-middle attack
PSTN .................................................. Public Switched Telephone Network
RA ........................................................ Technology-specific set of roaming authentication protocols
ra ∈ RA ............................................... Roaming authentication protocol
\((R)A\) ................................................. Technology-specific set of roaming and home authentication protocols
\((r)a \in (R)A \) ........................................ Roaming or home authentication protocol
RSS ..................................................... Technology-specific set of roaming security suites
\(RSS_{X \text{--allow}} \) ....... Set of roaming security suites $X$ allows to be used upon roaming.
RKA ................................................ Technology-specific set of roaming key-agreement protocols
rka ...................................................... Roaming key-agreement protocol
\((R)KA \) ............................................ Technology-specific set of roaming or home key-agreement protocols
\((r)ka \) ............................................... Roaming or home key-agreement protocol
\( S_0 \) .................................................. Initial Security Context
\( S_k, 1 \leq k \leq h \) .............................. Security-context transferred on a $k$-th order handover
SC ..................................................... Security Center
SRC ................................................................... Source network
SS ..................................................... Technology-specific set of security suites
$ss \in SS$ ............................................................... Security suite
$SS_{X-allow}$ ........................................... Set of security suites expressing the policies of $X$
$ssh_k$ .................................................. Security suite history on a $k$-th-order handover
SCT .......................................................... Security-Context Transfer
UMTS ........................................ Universal Mobile Telecommunications System
WLAN .................................................... Wireless Local Area Network
$\mathbb{Z}_n$ ........................................................ Group of residues modulo $n$
$\mathbb{Z}_n^*$ ................................................ Multiplicative group of residues modulo $n$

Notations Used in Figures
Message 1 .............. Message sent from A to C, forwarded by B without any changes
Message 2 .............. Message sent from A directly to C, without B’s involvement
Message 3 ........ Message sent from A to C, forwarded by B after processing or completion

Figure 1: Notations Used in Figures

Terms specific to Part III
A3/A8 .................................................. GSM authentication and key-generation algorithms
A5 .......................................................... GSM encryption algorithms
AuC ........................................................ Authentication Center
AUTN ..................................................... Authentication Token
BSC ........................................................ Base Station Controller
BTS ................................................... Base Transceiver Station (GSM)
CK .......................................................... UMTS encryption key
### Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GERAN</td>
<td>GSM EDGE Radio Access Network</td>
</tr>
<tr>
<td>HLR</td>
<td>Home Location Register</td>
</tr>
<tr>
<td>IK</td>
<td>UMTS integrity protection key</td>
</tr>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber Identity</td>
</tr>
<tr>
<td>$K_c$</td>
<td>GSM encryption key</td>
</tr>
<tr>
<td>MSC</td>
<td>Mobile Switching Center</td>
</tr>
<tr>
<td>NodeB</td>
<td>Base Transceiver Station (UMTS)</td>
</tr>
<tr>
<td>$RAND_G$</td>
<td>GSM authentication challenge</td>
</tr>
<tr>
<td>$RAND_U$</td>
<td>UMTS authentication challenge</td>
</tr>
<tr>
<td>$RES_G$</td>
<td>GSM authentication response</td>
</tr>
<tr>
<td>$RES_U$</td>
<td>UMTS authentication challenge</td>
</tr>
<tr>
<td>RNC</td>
<td>Radio Network Controller</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SQN</td>
<td>Sequence Number</td>
</tr>
<tr>
<td>TMSI</td>
<td>Temporal Mobile Subscriber Identity</td>
</tr>
<tr>
<td>UEA</td>
<td>UMTS Encryption Algorithm</td>
</tr>
<tr>
<td>UIA</td>
<td>UMTS Integrity Algorithm</td>
</tr>
<tr>
<td>USIM</td>
<td>Universal Subscriber Identity Module</td>
</tr>
<tr>
<td>UTRAN</td>
<td>UMTS Terrestrial Radio Access Network</td>
</tr>
<tr>
<td>VLR</td>
<td>Visitor Location Register</td>
</tr>
</tbody>
</table>

### Terms specific to Part IV

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>CARD</td>
<td>Candidate Access Router Discovery</td>
</tr>
<tr>
<td>CCMP</td>
<td>Counter Mode with CBC-MAC Protocol</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CTD</td>
<td>Context Transfer Data</td>
</tr>
<tr>
<td>CTAR</td>
<td>Context Transfer Activate Request</td>
</tr>
<tr>
<td>CT-Request</td>
<td>Context Transfer Request</td>
</tr>
<tr>
<td>CXTP</td>
<td>Context Transfer Protocol</td>
</tr>
<tr>
<td>DS</td>
<td>Distribution System</td>
</tr>
<tr>
<td>GTK</td>
<td>Group Transient Key</td>
</tr>
<tr>
<td>KCK</td>
<td>Key Confirmation Key</td>
</tr>
<tr>
<td>KEK</td>
<td>Key Encryption Key</td>
</tr>
<tr>
<td>PMK</td>
<td>Pairwise Master Key</td>
</tr>
</tbody>
</table>
PRNG ............................................. Pseudo Random Number Generator
PSK ................................................................. Pre-Shared Key
TKIP .................................................. Temporal Key Integrity Protocol
TK .......................................................... Temporal Key
WEP ...................................................... Wired Equivalent Privacy
WPA ........................................................ Wi-Fi-Protected Access
WPA2 ..................................................... IEEE 802.11i
WISP ...................................................... Wireless Internet Service Provider
UAM ..................................................... Universal Access Method
Introduction

The goal of this thesis is to model the security challenges imposed on infrastructure-based wireless access networks by roaming and handover procedures between different network providers and across different technologies, to analyze current solutions within this model, and to develop new security solutions.

Roaming, Handover and Security

In recent years, wireless technology has become an integral part of state-of-the-art networking and the use of various wireless devices has become part of our everyday life: in August 2005 85% of Germans carried cell phones [86], most PDAs currently have a wireless interface, and it is almost impossible to buy a new laptop that does not come with a built-in radio. Mobile phone operators offer mobile telephony and data services to their customers, companies and universities allow their employees or students to access the local networks wirelessly, hotspot operators offer internet connectivity in public areas such as airports and coffee shops, and in more and more private homes, computers, printers and fax machines are connected wirelessly. Some of the most well-known and most widely used wireless access technologies are the mobile telecommunication standards GSM, UMTS, and CDMA2000, the data service enhancement of GSM called GPRS, as well as the local area networking technologies IEEE 802.11 and Bluetooth.

A growing number of technologies and providers as well as the user’s increasing need and desire to be connected and reachable at all times, call for solutions which allow for the inter-operation of the different technologies and providers.

Roaming and handover procedures are two forms of inter-operation that aim to support user mobility across different providers and technologies. Before we proceed, we must briefly explain these terms.

In a mobile phone network, a user typically subscribes to one mobile phone operator, for example T-Mobile®. This subscription allows him to place and receive phone calls over the T-Mobile® network in a certain geographical area, like Germany. To support the mobility of its subscribers, T-Mobile® enters into roaming agreements with other mobile phone operators. If T-Mobile® and, e.g., the France Télécom network Orange® have a roaming agreement, a T-Mobile® subscriber can place and receive phone calls while in France using the Orange® network. Thus the roaming agreement between the two operators and the

\footnotesize{1T-Mobile® is a registered trademark of Deutsche Telekom AG.}
\footnotesize{2Orange® is a registered trademark of France Télécom.}
Introduction

associated roaming procedure allows the user to access the Orange® network without prior subscription to this network operator. This network access includes being able to receive incoming phone calls as well as being able to place phone calls. On the network side, roaming thus includes updating the location information for a roaming user, as well as re-routing incoming user traffic to the user’s new point of network attachment.

With roaming service alone, the call a traveling user places while still in Germany will drop as soon as the user crosses the border and the T-Mobile® network is no longer available. Here, handover procedures come in. If T-Mobile® and Orange® had a handover agreement, then the associated handover procedure would allow the user to continue his phone call without interruption over the Orange® network as soon as the T-Mobile® network is no longer reachable. On the network side, a handover procedure thus requires incoming and outgoing traffic on an ongoing connection to be re-routed to the user’s new point of network attachment without causing an interruption. This places tight efficiency requirements on handover procedures, which may also result in different re-routing paths than occur in the roaming case.

While best known from mobile phone networks, inter-provider roaming and handover procedures have found their place in other commercial wireless access networks, such as the wireless LAN networks of hotspot providers. Non-commercial inter-provider roaming scenarios include roaming between university or company-owned WLAN networks that are administered by different entities.

Previously, wireless devices were equipped with one technology only, thus limiting inter-operation to providers supporting the same technology. A recent trend, however, is to integrate more than one wireless communication interface in a single mobile device (see, e.g., [32]), thus allowing the user to benefit from the advantages of different networking technologies. For example, a user with a PDA supporting both GPRS and 802.11 cannot only use a local area network with its high data rates, but may also benefit from the large coverage area of a GPRS network. Inter-system roaming procedures allow users pre-registered for a network of one network technology to use a network of another technology without prior registration for the latter one. Inter-system handover procedures allow such users to maintain an ongoing connection while changing from one access network technology to another.

With the advantages of mobility support and the technical challenges of supporting handover and roaming come just as many challenges in providing secure solutions for roaming and handover between different technologies and providers. Currently, most providers restrict the use of their networks to pre-registered users. Through the pre-registration process, the user and a dedicated provider establish a trust relationship and exchange credentials such as cryptographic keys, which enable a secured network access. The network for which a user is pre-registered is typically referred to as his home network. Any other network is called a foreign network.

The security challenge on inter-provider roaming is enabling a mobile device and a foreign network to authenticate each other, negotiating security mechanisms, and establishing cryptographic keys to secure the network access without any prior direct trust relationship.
The additional security challenge on inter-system roaming is to enable the usage of the same credentials on authentication across different access technologies.

In order to offer continuous use of services, handover procedures have to be fast for several reasons. First, real-time services like voice or video connections are sensitive to short disruptions. Second, users are sensitive to disruptions. Third and finally, during a handover procedure a user typically moves out of the range of his currently serving network such that a handover procedure has to be completed before the user looses connection to his original point of network attachment. The security challenge on inter-provider handover is to enable a secure network access fast enough to allow for uninterrupted use of ongoing connections. This secure access includes (indirect or direct) mutual authentication between a mobile device and the destination network as well as establishing cryptographic keys and negotiating cryptographic mechanisms to use after handover.

Different providers and users have varying requirements, and different technologies have varying security capabilities (e.g., encryption or authentication methods, key sizes). Consequently, handover and roaming across different technologies and providers means crossing domains that are not equally well protected. This arises the question, how the security policies of providers and users during handover and roaming should be taken into account.

In this thesis we model the security challenges imposed on infrastructure-based wireless access networks by roaming and handover procedures between different network providers and across different technologies. We analyze current solutions within this model, and develop new security solutions.

In particular, we develop a new approach for public-key cryptography-based inter-provider roaming and suggest a new roaming authentication protocol EAP-TLS that implements the new approach in the context of WLANs. We present the first formal model for subsequent context transfers on inter-provider and inter-system handover. We provide a thorough threat analysis of inter-provider and inter-system handover procedures from which we derive new security requirements. We develop a new security solution for inter-provider and inter-system handover that enhances state-of-the-art security-context-transfer-based solutions and meets the newly defined security requirements. We exemplify this new handover approach in the context of wireless local area networks. Moreover, as a case study for inter-system handover, we present the first comprehensive security analysis of the handover and roaming procedures standardized for the inter-operation of GSM and UMTS and compare them to the security requirements and design goals we derived. We show that these procedures are vulnerable to a variety of attacks and present security solutions that can be used on top of the current security architectures of GSM and UMTS. The above summarized contributions of this thesis are detailed in the following.

Contributions

Roaming

- The classification, modeling, and discussion of current security solutions for both inter-provider and inter-system roaming. As opposed to other overviews on roaming procedures, such as [178, 24], our description is independent of any particular wire-
less access technology. Roaming procedures are classified according to the amount of control a home network retains and according to where the cryptographic keys that protect the network access after successful authentication are generated. Other technology-independent work on roaming procedures (e.g., [29, 156]) does not address how the security mechanisms used between a foreign network and mobile device should be negotiated. As roaming typically requires the home network’s authorization, it is in the home network’s interest that the mobile device and the foreign network adequately secure their connection. We explicitly address this problem and show that the home network of a mobile device should actively participate in the security-mechanism negotiation.

- The development of a new approach to public-key-based inter-provider roaming. Our approach enables a home network to control each roaming instance. By means of secret-sharing techniques this approach solves the two most common shortcomings of state-of-the-art roaming solutions. First, as opposed to [156, 18, 189, 9, 87], it does not require the transfer of secret session-key material from the home network to the network to which a mobile device roams. Second, as opposed to other solutions that use public-key certificates to authenticate the network to the mobile device (e.g., [29, 81]), the new approach spares a mobile device from any certificate validation.

- The development and security analysis of EAP-TLS-KS, a protocol that enhances EAP-TLS to support the new secret-sharing approach for inter-provider roaming between IEEE 802.11i protected WLANs. This part of the thesis is joint work with S. Wetzel and J. Cordasco and has been published in [121].

- A security analysis of the roaming procedures between GSM and UMTS. The most important result of the analysis is that UMTS networks that inter-operate with GSM are vulnerable to a man-in-the-middle attack. This part of the thesis is joint work with S. Wetzel and has been published in [122]. After publication, this attack was discussed by the standardization organization 3GPP [3] and included in the vulnerability and enhancement study [2]. EAP-TLS-KS is a public-key-based approach and, consequently, cannot be used to enhance GSM-UMTS inter-operation without major changes to the standardized security architectures of these wireless technologies. The solution we have suggested in [122] is specifically designed to be applicable on top of the standardized symmetric-key-based architectures.

Handover

- The classification, modeling, and discussion of current security solutions for inter-provider and inter-system handover procedures. Security solutions are classified into methods that use a full (roaming) authentication and key agreement between a network and a mobile device prior to or during handover and methods that use security-context transfer. Solutions that use security-context transfer (SCT) reuse already established cryptographic keys to derive and transfer new keys to the destination
network. The security-context transfer implicitly authenticates the mobile device to
the destination network. We discuss these security solutions and their applicability
for technologies supporting different types of handover procedures. As opposed to
previous work on security-context transfer solutions for inter-provider roaming (e.g.,
[186, 177, 74, 185, 55, 162]), we explicitly model the subsequent context transfer arising
from several subsequent handover.

• A detailed analysis of potential threats against security-context transfer methods on
inter-provider as well as inter-system handover. In this threat analysis, we describe
various attacks that can be enabled by security-context transfer in combination with
weaknesses in one of the previously serving networks or the currently serving one.
Some of these attacks arise from the dependencies of the cryptographic keys used
before and after handover. Some arise from weaknesses in the methods used to nego-
tiate security mechanisms, and others are mainly enabled by weaknesses in the
used security mechanisms themselves. Although the use of security-context trans-
fer to accelerate inter-provider and inter-system handover is widely discussed (e.g.,
[186, 177, 74, 185, 129, 55, 75, 162]), the inherent threats of SCT have not been ade-
quately treated in literature yet [75, 111]. We close this gap by our detailed analysis.

• A new history-enriched, policy-based approach to inter-provider and inter-system han-
dover that explicitly addresses subsequent handover, as well as handover of roaming
users. This new approach protects against attacks arising from the use of weak secu-

rity mechanisms before handover by guaranteeing forward secrecy of the transferred
keys. Attacks enabled by the use of weak mechanisms after handover, are prevented by
enforcing the mobile device’s, the handover controlling network’s, and the destination
network’s policy upon the security-mechanism negotiation and protecting the negoti-
ation against bidding-down attacks. By adequately protecting the handover message
exchange between the mobile device and the networks, attacks that use manipulation
or interception of handover-related message are prevented. Our new approach
enhances security-context transfer with a key history that contains information on
how the cryptographic keys in the security context were generated and how keys de-
derived from them were used so far. This approach enables users and providers to base
their handover decisions on the history of the transferred cryptographic keys. Con-
sequently, users as well as providers can prevent critical handover situations and yet
inter-operate with providers and users that support untrusted security mechanisms.
The ability to reject a handover on a per-case basis allows users and providers to
protect against a third set of attacks. Handover decisions are thus more flexible and
respect the security policies of both users and providers. This part of the thesis is
joint work with S. Wetzel and will be published in [124].

• The development and analysis of the history-enriched, policy-based security-context
transfer approach applied to IEEE 802.11 WLANs. We show how our new security-
context transfer approach can be integrated into the context transfer protocol sug-
gested in [111]. Moreover, we show how the concepts underlying our new roaming
solution EAP-TLS-KS can be used to generate fresh cryptographic keys between a
foreign network and a mobile device during handover.

- A security analysis of the inter-system handover procedures between GSM and UMTS.
  In particular, we detail how GSM vulnerabilities like weak encryption mechanisms or
  fake base stations can spread to UMTS by means of handover procedures. This part
  of the thesis has been published together with S. Wetzel in [123]. These vulnerabilities
  were discussed by 3GPP in [3] and lead to some new recommendations for GSM opera-
  tors [4]. The solutions we presented in [123] built on the existing security architectures
  for GSM and UMTS. However, our history-enriched, policy-based solutions could be
  used to enhance the inter-operation between GSM and UMTS requiring changes to
  the inter-provider communication only.

Outline. The remainder of this thesis is organized in four parts. In the first two parts,
theoretical in nature, we model wireless access networks, roaming, and handover and develop
and analyze our new solutions in a technology independent way. In the last two parts, our
new solutions are applied to inter-provider roaming and handover in WLANs and inter-
system handover between GSM and UMTS.

We model wireless access networks in Chapter 1 and roaming in Chapter 2 of Part I.

The handover model is presented in Part II. This part starts off with a model of different
types of handover procedures, the security challenge, and a discussion of potential security
solutions in Chapter 3. In Chapter 4, we analyze the potential threats arising from security
solutions based on security-context transfer. We present our own history-enriched, policy-
based security-context transfer in Chapter 5. Part II ends with an extension of our inter-
provider solution to the inter-system handover case and an overview on other related issues.

In Part III, we study inter-provider and inter-system roaming and handover between
GSM and UMTS. Chapter 7 briefly describes the system model and the security architectures
of GSM and UMTS, including inter-provider roaming within GSM and UMTS. The
inter-system roaming procedures are described and analyzed in Chapter 8. The inter-system
handover procedures are described and analyzed in Chapter 9.

Finally, in Part IV, we apply our new roaming and handover solutions to the WLAN case.
We start with a brief overview on the WLAN system model and the new security architecture
802.11i for WLAN in Chapter 10. In Chapter 11, we present our new secret-sharing-based
roaming solution EAP-TLS-KS. The adaption of the history-enriched, policy-based security-
context transfer solution and our new key-agreement method using secret-sharing techniques
is presented in Chapter 12. The thesis ends with a Conclusion.

Each part in this thesis begins with a brief overview of how this part is related to other
parts. Each chapter begins with an introduction that motivates its content, details the
contributions of the chapter, and provides a chapter outline. Each chapter ends with a
detailed discussion of the related work and a conclusion that summarizes the content of the
chapter and provides input for future directions of research.

Reading Instructions. The second part of this thesis requires many of the terms intro-
duced in Part I and can thus not easily be read outside this context. However, Part III
and Part IV are written to be self-contained such that they can be read without deeper understanding of the theoretical background provided in Part I and II.
Part I

Wireless Access Networks and Roaming
Part I in the General Context
In Chapter 1, we model wireless access networks with respect to their system and security architecture. In Chapter 2, we model and classify roaming authentication and key-agreement protocols between wireless access networks. Moreover, we introduce a new approach to public-key-based inter-provider roaming in Section 2.2. Part I founds the bases for Part II, in which we model inter-provider handover procedures for mobile devices. In Chapter 11, we detail the new roaming approach introduced in Section 2.2 in the context of WLANs. The notations and terms introduced in Part I are used throughout the remainder of this work.
Chapter 1

System Model and Security Model

Wireless networks come in two different forms: infrastructure-based networks and infrastructure-less networks, so called ad hoc networks. In an ad hoc network, mobile devices interconnect without the help of a fixed infrastructure. This means that the mobile devices detect each other, establish connections, and route traffic between themselves. In an infrastructure-based network, these tasks are left to a fixed, wired or wireless infrastructure. Throughout this work, we focus on infrastructure-based wireless access networks.

In this chapter, we model infrastructure-based wireless networks (referred to as wireless access networks in the remainder of this work) as to their components and security architecture in a technology-independent way. We identify the entities user, mobile device, network provider, home network provider, and service provider, as well as the components of wireless access networks, and describe their relationships. Furthermore, we determine the security goals of users and providers in a wireless access network and model the security mechanisms used to protect them.

In particular, we define mutual authentication and key-agreement protocols between a mobile device and its home network, key-establishment processes by which data-protection keys are derived from master session keys, as well as encryption and integrity-protection mechanisms. We model the security-related part of the establishment of a connection between a mobile device and its home network.

Our model is based on the components and security architectures of GSM, UMTS, IEEE 802.11, and CDMA2000 [62, 9, 91, 93, 153], but it also complies to the security architectures of Bluetooth and IEEE 802.16 [96, 100]. The goal of our model is to found the basis of our futher enhancements to roaming and handover procedures.

In addition to the general model, we describe various methods a mobile device and its home network can use to negotiate security mechanisms based on their security policies. State-of-the-art technologies use very simple negotiation methods (see Method 1 to Method 3 in Section 1.3) that do not enable users or providers to express preferences with respect to the security mechanisms they allow to be used. This contradicts the observation that users and providers typically deem different security mechanisms to provide different protection levels and aim to use security mechanisms that provide as high a protection level as possible.
We present and discuss one already existing negotiation method that take the preferences of both users and providers into account (see Method 4 in Section 1.3). This method was previously suggested in the context of reconciliation of security policies in general [179] and builds on the assumption that the preferences of one negotiating party outweigh the other. Furthermore, we introduce a new negotiation method that takes the preferences of both parties with equal weights into account and is resistant against some manipulations by the negotiating parties (see Method 5 in Section 1.3).

Outline. In Section 1.1, we describe the components of a wireless access network. Section 1.2 starts with the specification of the security goals of wireless access network providers and users in Section 1.2.1. We then model the pre-registration process between a user and his home provider and the security mechanisms used between the two in order to protect their security goals in Section 1.2.2. In Section 1.2.3, we summarize our assumptions on the security policies of users and providers. In Section 1.2.4, we describe how the different security mechanisms are used between a mobile device and its home network on connection establishment. Finally, in Section 1.3, we model existing and introduce new methods to negotiate security mechanisms between a mobile device and its home network.

1.1 System Model

In a wireless infrastructure network, a mobile device (MD) like a mobile phone or a laptop connects to a fixed network access point (NAP) over a radio connection. Examples for network access points are base transceiver stations in wireless phone networks or access points in a wireless LAN. Every NAP has a transmission range. A mobile device can connect to a NAP only if it is in the NAP’s range. In the simplest case, a wireless access network consists of only one network access point that is itself connected to a wired network. A single network access point can cover only the area of its own transmission range. To be able to cover a wider area, more than one network access point has to be used. These network access points themselves are connected, either over radio or wired, to a backbone network.

In some wireless technologies, this backbone network is hierarchically structured. Several network access points are connected to a more central network component, several of these are connected to an even more central network component, and so on. Mobile phone networks are typical examples for infrastructure networks with a hierarchical backbone. Figure 1.1 illustrates a GSM network as an example for a wireless network with a hierarchical backbone.

In other wireless technologies, the backbone network is flat. For example, in an IEEE 802.11 WLAN, several network access points can be directly interconnected by a LAN technology like Ethernet and connected to the Internet via a router. This is illustrated in Figure 1.2.

1Depending on the wireless technology and the environment in which it is used, the coverage area of a network access point can range from several meters up to several dozen kilometers (e.g., 3-35 km in GSM, 10-200m in WLAN).
1.1 System Model

For the remainder of this work, the detailed structure of the backbone network is not taken into account. A wireless access network can therefore simply be described by Figure 1.3. A wireless access network consists of several network access points that are connected to a common backbone network. This backbone network itself can be connected to other wired networks, like a local area network, the Internet, or the PSTN.\(^2\)

A wireless access technology typically specifies only the first two layers of the ISO/OSI reference model, namely the physical (PHYS) layer and the medium access control (MAC) layer. It allows for the integration of different network layers and often even different MAC and PHYS layers on the backbone network. The network access points (NAPs) act as a bridge between the wireless MAC layer and the MAC layer of the backbone network. Figure 1.4 illustrates this.

A wireless access network is operated by a network provider. A provider is a public or private entity that offers connectivity to a mobile device via a wireless access network. One widespread type of network providers are commercial providers like operators of mobile phone networks or WLAN hotspot providers. But providers can also be non-commercial, like companies or universities that offer wireless network access in addition to wired access to their employees or students and staff.

Another type of non-commercial provider is non-profit organizations that offer free wireless access to everyone in public places like libraries, parks, or coffee shops. An example for a network like this is the freely available wireless LAN network the Bryant Park Restoration

\(^2\)Public Switched Telephone Network.
Corporation offers in New York City [42]. On a larger scale, the city of San Francisco is currently planning to offer free WLAN access to all its citizens. The Mayor of San Francisco even called wireless access a “fundamental right” [22]. Although these freely accessible networks exist, they are not the focus of this work. Instead, we only consider commercial and non-commercial network providers that want to restrict access to their network to a certain set of users.

Every user is pre-registered for one wireless access network, his home network operated by one provider, his home provider. In the registration process, a user and the network provider establish a trust relationship and exchange credentials and cryptographic keys. The registration of a user is with respect to a certain identity that represents the pre-registered user upon future network access. Examples for an identity like this are a username assigned to a user by its home provider, a worldwide unique identifier like an the International Mobile Subscriber Identity (IMSI) used in GSM and UMTS networks or an identifier of the form “user@domainname”. Other examples are IP or MAC addresses. The credential for a user is issued for the user’s pre-registration identity. In the following we will refer to the pre-registration identity as the user’s identity. Depending on the type of credential and pre-registration identity issued for a user, the pre-registration may be independent of or be bound to a certain MD. Credentials issued for the user can be stored on a particular MD or be usable on any MD. In the following we do not differentiate between these two cases, that is, we assume that each pre-registered user is issued a credential and uses this credential in
connection with one particular MD. As a consequence we do not differentiate between a pre-registered user and its (pre-registered) MD. In particular, we are not interested in securing the interface between users and MDs and do not differentiate between the pre-registered user, using his MD, the owner of MD, or a person that is authorized by the pre-registered user to use his MD.

A user uses his MD to connect to a wireless network in order to use a service. For example, a user may access the network in order to connect to a local resource in the backbone of the wireless access network, like a printer. Other examples of services are placing and receiving phone calls or establishing IP-connectivity. The services a user can use via a wireless access network are provided by service providers.

Figure 1.5 illustrates different entities—the user, the home network provider, the foreign network provider, the service provider, the user’s mobile device, and the wireless access network—as well as their relations to each other. A service provider can coincide with the home network provider (e.g., access to a local printer, IP-connectivity, voice calls) or be different from the home provider (e.g., location-based services).

1.2 Security Model

1.2.1 Security Goals and their Protection

This section describes the security goals of providers and users of a wireless access network and compares them to the ones typically postulated for wired networks. Aside from anonymity (Section 1.2.1.4), the security goals are the same for wired and wireless access networks. However, the broadcast nature of a wireless access network and the easy accessibility of the air interface make it easier to violate and harder to protect these goals.

1.2.1.1 Authentication and Authorization

Protect against unauthorized users: The provider of a wireless access network wants to restrict access to its network to a certain set of authorized users.

In commercial wireless access networks like mobile phone networks or wireless hotspot networks, only those users who pay for the service should be able to access the network.
Likewise, a company’s wireless access network should only be accessible by employees or a certain group of employees. Networks are protected from unauthorized users by an authentication procedure. By means of this procedure, the user proves his identity to the network. Once the user is identified, the network can determine whether the user is authorized to access the network.

In a wired network, devices like personal computers or terminals are connected to the network via sockets. These sockets and devices are physically protected by the buildings and rooms they are in. Access to the network thus requires physical access to the sockets and/or the user devices connected to the sockets. This physical protection is missing in wireless access networks: everyone has physical access to the air interface. In the “classical” case of a wired network, a user authenticates himself to the network via a fix device that is connected to the network. Mobile devices like laptops are more and more commonly used to access wired networks. Lately, the problem of authenticating users with their own mobile device when it is plugged into a socket of the wired network has become more widely recognized [19].

**Protect against unauthorized networks:** A user wants to be protected from accessing unauthorized networks.

An attacker can try to make a mobile device connect to a “fake” network access point that is controlled by the attacker. Once a mobile device is connected to a fake network access point, the data confidentiality of user data and of user-related data like its identity can be violated.

To protect against unauthorized networks, the network is authenticated by the mobile
device and the mobile device checks its authorization to offer network access to the user. Examples for wireless access network technologies in which a missing network authentication allows for fake network access points are GSM and WLANs. The corresponding attack for GSM is briefly described in Section 9.4.1.2, while the attack on UAM-protected WLANs can be found in [178].

In a wired network, the physical location of the network access device or sockets already give the user some confidence in the identity of the network he is currently accessing. In the wireless case, everyone has physical access to the air interface and can broadcast beacons and try to make mobile devices connect to fake network access points. With respect to network authentication, a wireless access network has more similarities to accessing a wired network from a remote machine than from a machine that is directly connected to the network in question. On remote network access, a user also has to make sure he is accessing the network he intends to access before sending any sensitive user data over the network.

A mechanism that supports both user and network authentication is called a mutual authentication mechanism. Most current wireless access technologies support mutual authentication between user and home network.

1.2.1.2 Confidentiality

**Protect the confidentiality of data and control traffic:** Users and providers want to protect the confidentiality of the data and control traffic on the air interface between a mobile device and a network access point.

In wireless access networks there are two different types of traffic exchanged between a mobile device and the network: data traffic and control traffic. Control traffic is all the information exchanged between a mobile device and the network concerning the establishment and control of a connection and the management of the network. This includes, for example, authentication traffic and traffic related to the negotiation of security mechanisms. On the other hand, data traffic is the traffic sent and received by a mobile device as the result of the usage of a service. An example of data traffic is the voice data of a phone call placed or received over a mobile phone network. Another example is IP traffic related to a TCP session established over the wireless access network between the mobile device and a corresponding node at some location on the Internet.

---

3 The Universal Access Method (UAM) is the currently most authentication method used by hotspot provider [18].

4 Control messages informing all mobile devices within the range of a NAP of its presence, possibly along with additional status or configuration information.

5 Depending on the network technology, faking a network access point can be more or less costly: while a WLAN access point can be purchased starting from US$30, a GSM base station is hardly affordable for an attacker operating on his own. Moreover, broadcasting on the technology-dependent frequencies without anyone noticing can be hard to do if the corresponding frequencies are regulated. For example, this is the case for mobile phone technologies.

6 Transmission Control Protocol.
In mobile phone networks, different channels are used for data traffic and control traffic. In packet-switched networks, control and data traffic is exchanged over the same channel. In a wireless network all traffic between a mobile device and a network access point is broadcast on the air interface. An attacker with a network analyzer can thus easily intercept all traffic going back and forth between a network access point and associated mobile devices. To protect against an unauthorized third party getting access to control and data traffic on the air interface, the traffic is typically encrypted. Encrypting the control traffic protects, for example, the confidentiality of connection parameters and commands to switch the channel. It is important to note that some wireless access technologies do not support encryption of control traffic and only encrypt data traffic on the air interface [62, 9]. Encrypting data traffic on the air interface does not protect the confidentiality of the end-to-end connection between a user and a corresponding node. For example, if the voice data of a phone call is encrypted on the air interface, the plaintext of the voice data is available everywhere on the backbone network and the PSTN. End-to-end protection, if desired, thus has to be implemented on a higher layer and between the end points of the voice connection. Yet the encryption of the air interface protects the user data while it is particularly easy to intercept. Moreover, encrypting user data includes protecting the information contained in the header of higher-layer protocols and thus prevents traffic analysis for these higher layers. If, for example, data traffic is encrypted on the MAC layer, then the source and destination addresses and the payload of an IP packet are encrypted. Another advantage of encryption on the MAC layer is that independent of any higher-layer protocol, all data traffic is encrypted and not just the traffic on top of a certain protocol.

It is important to note that in order to be able to encrypt traffic between a mobile device and the network, the network and the mobile device have to negotiate an encryption mechanism and agree upon an encryption key. As long as the encryption mechanism and the encryption key are not “in place,” encryption is not possible.

1.2.1.3 Integrity Protection

Protect against manipulations of user and control traffic: Users and providers want to notice any manipulation, such as deletion, insertion, or reordering of user and control traffic on the air interface between a mobile device and a network access point.

An attacker that is able to change the control traffic between a mobile device and the network can interfere with the establishment and maintenance of the connection between a mobile device and the network. Unauthorized changes to control and data traffic also violate the authenticity of traffic sent by the network to the mobile device and vice-versa. If an attacker can make unauthorized changes, the mobile device and the network cannot be sure that they communicate with each other. It is important to note, that encrypting data or signaling traffic does not protect integrity as the network and the mobile device do not have any means of detecting whether decrypted plaintext is correct.

In order to protect against manipulations, user and control traffic on the air interface is
1.2 Security Model

integrity-protected.\(^7\) Integrity protection mechanisms cannot prevent alteration or insertion of traffic, but they can help the mobile device, or the network, to notice manipulations like this and react by discarding the traffic or even dropping the connection.

1.2.1.4 Anonymity

**Protection of the confidentiality of the user identity:** Users want to protect their identity from being revealed to anyone else but authorized networks.

This goal is postulated for some wireless access networks and refers to the confidentiality of the user’s identity on the air interface. A user’s identity should not be revealed to anyone but authorized networks (in some cases even to no one else but the home network) in order to protect the user from being localized and traced by an attacker intercepting traffic on the air interface. To anyone else, only the fact that some mobile device is currently connected to a given wireless access network should be revealed. It should be impossible to relate this MD to the pre-registered identity of the user or relate MD’s current connection to any previous one.

Protecting the confidentiality of a user’s identity is difficult. The wireless access network has to obtain the user’s identity in order to authenticate the user. Yet the user’s identity should not be sent over the radio interface in the clear. Encrypting the user’s identity, on the other hand, is only possible if the encryption key and mechanism already have been agreed upon. Nevertheless, in order to know which key and mechanism to use, the network has to gain knowledge of the user’s identity. Although the anonymity problem is out of scope of this work, it should be noted that recently some interesting solutions have been suggested for anonymity on inter-provider roaming (e.g., \([21, 188]\)). These are briefly discussed in Section 2.4.4.

1.2.2 Security Mechanisms

As motivated in the last section, wireless access networks should use a mutual authentication protocol to protect the network from unauthorized users and the user from unauthorized networks. This mutual authentication is based on secret and public information bound to a (pre-registered) user’s identity or a home provider’s identity. A user and his home provider exchange these credentials in a registration process.

The traffic on the air interface between a user and the network should be integrity-protected and encrypted. For this purpose, the network and the mobile device have to agree upon encryption and integrity-protection keys and mechanisms.

In this section, the terms registration process, authentication protocol, encryption mechanism, integrity-protection mechanism, key agreement, and key establishment are defined. These definitions are in accordance with those in \([120]\). In addition, the components involved in these security mechanisms in a wireless network are identified.

\(^7\)Note that older mobile phone technologies often do not support integrity protection, and some newer technologies only protect control traffic against manipulations.
1.2.2.1 Registration Process

Each user has a dedicated relationship with one wireless access network provider, its home provider. In a registration process, the home provider and the user exchange credentials (i.e., secret or public information bound to the user’s identity and to the identity of the home network). As we assume a one-to-one relationship between a pre-registered user and his mobile device, we will in the following speak of the credential issued for a pre-registered mobile device. The pre-shared credentials allow HN and the mobile device of a pre-registered user to mutually authenticate each other when the pre-registered MD requests access to HN. Depending on the credential type, the mutual authentication between MD and HN may require interaction with the user, as, e.g., in the case of a username/password combination, where the password is kept in the user’s mind, rather than being stored on his mobile device. To simplify notations we will however omit this fact throughout the rest of this work and simply speak of a mutual authentication between a mobile device and its home network.

On the network side the credentials are stored in database called Security Center (SC) here. The most widely used credential types in wireless access networks are:

1. **Shared secret key**: A (long-term) secret key is shared between the MD and the security center of HN (SC\textsubscript{HN}). MD stores its HN’s identity along with the shared key and SC\textsubscript{HN} stores the user’s identity along with the shared key.

2. **Public-key certificates**: MD and HN are each in possession of a public/secret key pair. Both have public key certificates for their own public keys. The certificate of MD is issued for the user’s identity. These certificates are signed by a Certification Authority (CA) with a secret key of CA. MD and SC\textsubscript{HN} store the public key certificate corresponding to CA’s private signature key.

3. **Public-key certificates mixed with a username/password combination**: The home network has a public/secret-key pair and a certificate signed by a CA. MD stores the public-key certificate corresponding to CA’s private signature key. The home provider issues a username (as user identity) and password for the pre-registered user and stores this combination securely in SC\textsubscript{HN}.

The registration process itself can differ greatly depending on the type of home provider. For example, a user may be identified by means of a passport, sign a contract and receive hardware, or install software on his MD as in the case of a user subscribing to a mobile phone operator. Registering may, however, also consist of walking into the system administrator’s office of a company, being identified, and then entering a preliminary password into the system administrator’s console or being handed a chip card with a public-key certificate and the corresponding private key. In other application areas, a user may pre-register with a network provider over a web-interface using, for example, his credit card number as credential. Whatever form the actual registration process takes, the result of the registration

---

\*Passwords are typically not stored in the clear. Instead, hash values of the passwords are stored.
is that MD and its HN have established credentials that subsequently allow them to mutually authenticate each other using a common authentication protocol.

The registration may additionally include an exchange of security policies regarding the security mechanisms to be used to secure the network access between MD and its HN. It may also include a registration for certain services offered by the home provider or other service providers over the home network. In the case of a commercial home provider, the user and the provider also agree upon charging and billing issues during the registration process.

In the remainder of this work, we refer to the user or his MD as being pre-registered to its home provider. We prefer this term over subscribed, as the latter is often associated with a commercial context. To additionally avoid confusion with the common usage of registered in the sense of being associated or having established a connection with a network, we use the term pre-registered.

1.2.2.2 Authentication Protocol

The authentication protocol terminates in a network component that is typically referred to as the authentication server (AS). An AS controls the access to a network for one or more NAPs. The authentication server needs access to user-related secret and public information that is securely stored in SC. An authentication protocol between MD and the authentication server of its HN (AS$_{HN}$) is based on one of the credential types introduced in Section 1.2.2.1 and has the following properties:

**Definition 1.2.1** A mutual authentication protocol ($a$) between MD and AS$_{HN}$ is a protocol between MD and AS$_{HN}$ whereby MD and AS$_{HN}$ are mutually assured of each other’s identity based on the credentials established during the registration process. Additionally, they are mutually assured that the other party has actually participated in the protocol.

It is important to note that two different authentication protocols specified for the same wireless access technology may be based on different types of credentials.

1.2.2.3 Key Agreement

In state-of-the-art wireless access network, symmetric encryption and integrity-protection mechanisms are used. These mechanisms require secret keys shared between the mobile device and the network. The keys have to be agreed upon directly or they must be derived from a secret master key agreed upon in a key-agreement protocol.

**Definition 1.2.2** A key-agreement protocol ($ka$) between MD and AS$_{HN}$ is a protocol between MD and AS$_{HN}$ whereby a shared master key $K$ becomes available only to MD

---

9In some wireless access networks, the functionality of an authentication server is implemented in the NAP.
This key is derived as a function of information ideally\textsuperscript{10} contributed by each party such that no party can predetermine the resulting value.

In many wireless technologies, authentication and key-agreement protocols are implemented as a unit and cannot be executed independently from each other. However, to be generic enough to capture future wireless technologies, we here consider that $a$ and $ka$ may be implemented independently. Thus MD and AS may agree upon new keys without authenticating each other. Vice-versa, MD and AS\textsubscript{HN} may newly authenticate each other without changing the keys previously agreed upon. The time since the last authentication and the lifetime of keys may thus differ.

### 1.2.2.4 Encryption Mechanism

In wireless access networks, the encryption and integrity-protection mechanisms are often implemented in the network access points. In some technologies, the encryption and integrity-protection endpoint (EIPE) is a network component to which more than one network access point are directly connected. This is, for example, the case in UMTS networks. Throughout this work, the EIPE is treated as a separate network component. We treat the case that the EIPEs coincide with the NAPs separately whenever necessary. If the NAPs in a wireless network are connected over a radio connection to the backbone network (as in many mobile telecommunication technologies), implementing the EIPE within a more central network component becomes a necessity, as otherwise the air interface between a NAP and the backbone becomes a serious threat to the confidentiality and the integrity of user and control traffic.

The encryption used in state-of-the-art wireless access network is typically a symmetric cipher.

**Definition 1.2.3** Let $n \in \mathbb{N}$ and $b \in \mathbb{N} \cup \{\infty\}$. A symmetric cipher on the alphabet $\{0, 1\}$ is a pair of functions $(E, D)$, $E : \{0, 1\}^n \times \{0, 1\}^b \mapsto \{0, 1\}^b$ and $D : \{0, 1\}^n \times \{0, 1\}^b \mapsto \{0, 1\}^b$ such that $E_{EK}(\cdot) := E(EK, \cdot)$ is invertible for every $EK \in \{0, 1\}^n$ and $D_{EK}(\cdot) := D(EK, \cdot) = E_{EK}^{-1}$. $n$ is called the **key length** of the symmetric cipher. If $b \in \mathbb{N}$, $b$ is called the **block length** of the symmetric cipher and the symmetric cipher itself is called a **block cipher**. If $b = \infty$ the cipher is called a **stream cipher**.

The **mode of operation** of a block cipher defines how subsequent plaintext blocks are encrypted. The simplest mode of operation is the so-called Electronic Code Book (ECB), where subsequent plaintext blocks are independently encrypted by the block cipher. Other modes of operation encrypt subsequent blocks dependent on previously encrypted blocks. Common examples (see [120] for details) are the Cipher-Block Chaining Mode (CBC), the Cipher Feedback Mode (CFB), and the Output Feedback Mode (OFB).

\textsuperscript{10}Some current key-agreement protocols do not fulfill this ideal property. Nevertheless, they are referred to as key-agreement protocols in the remainder of this work.
In most wireless technologies to date, symmetric stream ciphers are used. These ciphers can be interpreted as block ciphers with infinite block length. Stream ciphers encrypt plaintext of arbitrary length bit by bit, thus producing a stream of ciphertext bits.

Throughout this work, the term encryption mechanism is used in the following sense (see [120]):

**Definition 1.2.4** An encryption mechanism \((em)\) is a symmetric block cipher together with a mode of operation or a symmetric stream cipher.

In particular we assume that encryption mechanisms are symmetric.

### 1.2.2.5 Integrity-Protection Mechanism

Wireless technologies typically use a Message Authentication Code to protect the integrity of traffic between a mobile device and the network (e.g., UMTS, CDMA2000, TKIP part of 802.11i [9, 153, 93]). The integrity-protection endpoint on the network side coincides with the encryption endpoint and can thus either be implemented in the network access point or in a more central network component that serves more then one NAP. Some wireless technologies use an encrypted Modification Detection Code (MDC) for integrity protection and thus provide encryption and integrity protection in one (e.g., WLAN with WEP, CCMP part of 802.11i [93]).

The following definitions are taken from [120] and slightly modified to fit our notations.

**Definition 1.2.5** A message authentication code (MAC) is a family of hash functions \(h_{IK}\) parameterized by a secret Integrity Key (IK) with the following properties:

1. Given a key IK and a message \(x\), \(h_{IK}(x)\) is easy to compute.\(^{10}\)
2. \(h_{IK}\) maps an input \(x\) of arbitrary finite bit-length to an output \(h_{IK}(x)\) of fixed bit-length \(n\).
3. Given message/MAC-value pairs \((x_i, h_{IK}(x_i))\) \((i \geq 0)\), but not IK, it is computationally infeasible\(^{10}\) to compute any other message/MAC-value pair \((x, h_{IK}(x))\).

**Definition 1.2.6** A modification detection code (MDC) is a hash function\(^{10}\) that is pre-image resistant (i.e., it is computationally infeasible to compute \(x\) from \(h(x)\)) and 2nd pre-image resistant (i.e., given \((x, h(x))\) it is computationally infeasible to compute \(x'\) with \(h(x') = h(x)\)) and collision resistant (i.e., it is computationally infeasible to compute two pairs \((x, h(x))\) and \((x', h(x'))\) with \(h(x) = h(x')\)).

A MAC or MDC alone is not sufficient to detect a replay of previously recorded and correctly integrity-protected messages. This type of message insertion requires a special treatment, e.g., an additional counter or time stamp included in all messages. Throughout this work, the term integrity-protection mechanism is used in the following sense:

\(^{10}\)The precise definitions of these terms can be found in [120].
Definition 1.2.7 An integrity-protection mechanism (im) is either a MAC or a MDC together with an encryption mechanism em.\textsuperscript{11}

Wireless technologies that support integrity protection either protect control traffic only (e.g., UMTS, GPRS, CDMA2000) or both control traffic and data traffic (e.g., 802.11i-protected WLANs).

1.2.2.6 Key Transfer and Key Establishment

If AS and EIPE do not coincide the master key $K$ agreed upon in the key-agreement protocol has to be transferred from the authentication server to the EIPE.

Definition 1.2.8 A key-transfer process ($kt$) is a process whereby the master session key $K$ is securely transferred from AS to EIPE.

In some wireless technologies, the master session key generated during the key agreement is directly used for data protection. In other technologies, the data protection keys are derived in a key-establishment process from the master session key. Both possibilities are captured in the following definition:

Definition 1.2.9 A key-establishment process ($ke$) between EIPE and MD is a process whereby a secret data session key pair, here denoted by $(E_K, I_K)$, becomes available to MD and EIPE. The pair $(E_K, I_K)$ is derived as a function of the master key $K$ and optionally of other information contributed by MD and/or EIPE. If $(E_K, I_K)$ is derived from $K$ without additional information, the key-establishment process is called static. If additional information is contributed by MD and/or EIPE to derive $(E_K, I_K)$ from $K$, the key-establishment process is called dynamic.

For most wireless access technologies, more than one encryption and integrity-protection mechanism is standardized and used. These mechanisms may differ in their key length. We assume here that the key-establishment process generates encryption and integrity-protection keys of the right length for the selected encryption and integrity-protection mechanism.

Figure 1.6 and Figure 1.7 illustrate the security mechanisms and their endpoints on the network side in a wireless access network in case EIPE and NAP coincide and in case EIPE and NAP are different.

1.2.2.7 Security Mechanisms and the ISO/OSI Model

If the EIPE coincides with the NAP, the key establishment, encryption, and integrity protection are implemented on the MAC layer. In the other case, the encryption and integrity protection have to be implemented above the MAC layer. In case the encryption is implemented on the network layer or above, the wireless technology is potentially vulnerable to

\textsuperscript{11}In the second case the integrity-protection key $I_K$ coincides with the encryption key $E_K$ and includes replay protection.
address spoofing, as in this case higher-layer identifiers like an IP address of a mobile device might be revealed to an attacker eavesdropping on the air interface. If the encryption is implemented on the MAC layer, these higher-layer identifiers are encrypted along with the rest of the traffic. This is one of the advantages of implementing the encryption and integrity protection in the network access point. The authentication protocol, the key agreement, and the key transfer are typically implemented above the MAC layer, often even above the network layer, and in rare cases on the application layer. Figures 1.8 and 1.9 illustrate the relationship between the security mechanisms and the ISO/OSI layers on which they are typically implemented.
1.2.3 Security Policies

Throughout this work, we assume that for each wireless access technology, there are technology specific sets

\[ A = \text{Set of authentication protocols} \]
\[ KA = \text{Set of key-agreement protocols} \]
\[ KE = \text{Set of key-establishment processes} \]
\[ EM = \text{Set of encryption mechanisms} \]
\[ IM = \text{Set of integrity-protection mechanisms} \]

and a set \( SS \) of security-mechanism combinations that is a subset of the Cartesian product of the above sets:

\[ SS \subseteq A \times KA \times KE \times EM \times IM. \]

The elements \( ss \) of \( SS \) are referred to as security suites.\(^{13}\) The set \( SS \) of security suites is the set of all technologically possible and standard-wise allowed combinations of security mechanisms for the given technology. We assume that this set is known to every network provider and every user. Similarly, we define the set \( CS \) of cipher suites \( cs \) as

\[ CS := \{(ke, em, im) \mid \exists (a, ka) \in A \times KA \text{ such that } (a, ka, ke, em, im) \in SS\}. \]

A network provider may choose to implement and use only a subset of \( SS \) in the network he operates. This reflects the assumption that different providers may have different security policies. One provider might choose to allow only a specific security suite which he deems to guarantee adequate protection. Another one might implement and allow for the use of all standardized security suites.

A provider’s policy with respect to the security suites he allows to be used in his network \( N \) is expressed by a subset \( SS_{N-allow} \subset SS \). \( SS_{N-allow} \) includes all combinations of authentication protocols, key-agreement protocols, key-establishment processes and encryption and integrity-protection mechanisms network the provider allows to be used. The network components of a provider’s network support more than the allowed security suites as network component manufacturers will typically produce standard conform components that support all standardized security suites.

On the other hand, each mobile device has a certain set of security suites it supports. These security suites can come with the mobile device, with additional hardware the user receives from the home provider during the registration process, or as software the user installs on his mobile device. From the supported security suites, the user chooses a certain

\(^{12}\)Note that each authentication protocol is based on a certain type of credentials. The technology-specific set of authentication protocols thus indirectly defines a technology-specific set of credential types.

\(^{13}\)As a convention, we denote sets with capital letters and elements of a set with small letters (e.g, the set of authentication protocols \( A \) consists of elements \( a \)).
subset he allows to be used.\footnote{In some wireless access networks, users cannot configure the security suites they are willing to use. In this case, the users’ policies are ignored by the technology. Other technologies allow users to set any policy they want and consequently open up to vulnerabilities introduced by careless or uneducated users.} This subset is denoted by $SS_{MD-allow}$. It reflects the assumption that every user wants to achieve a minimal security level and disallows the security suites that he deems not to guarantee his minimal security level.

The security-policy expressions $SS_{MD-allow}$ and $SS_{HN-allow}$ alone do not indicate any preferences of a user or a network provider on the security suites they allow. These sets only express which security suites the provider or the user deems to guarantee a certain minimal security level he requires. Although this is the way security policies are expressed in many current wireless access networks (e.g., GSM, UMTS, CDMA2000 and 802.11i), ignoring the preferences of users and providers seems unnecessarily restricting: both the user and the network provider can prefer certain security suites above others and yet allow all of them. Consequently, the goal of users and providers is to negotiate the most preferred security suite they both support. We assume that the providers and the users each have a preference order $\preceq_{HN}$ and $\preceq_{MD}$ on their security-policies expressions $SS_{MD-allow}$ or $SS_{HN-allow}$. We assume here that these preference orders are total:

Definition 1.2.10 A \textit{total preference order} on a set $S$ is a relation $\leq$ on $S$ with the properties

1. \textit{(Antisymmetry)} If $s_1, s_2 \in S$ and $s_1 \leq s_2$ and $s_2 \leq s_1$ then $s_1 = s_2$
2. \textit{(Totalness)} If $s_1, s_2 \in S$ then $s_1 \leq s_2$ or $s_2 \leq s_1$
3. \textit{(Transitivity)} If $s_1, s_2, s_3 \in S$ and $s_1 \leq s_2$ and $s_2 \leq s_3$, then $s_1 \leq s_3$.

Note that in order to be able to establish a secure connection between them, MD and HN have to allow for at least one common security suite. MD and HN have to ensure this during pre-registration, as otherwise MD cannot access HN.

1.2.4 Connection Establishment

On connection establishment, MD and NAP$_{HN}$ first establish a radio link, which may include the allocation of a channel or other local wireless resources, possibly including a minimum level of service or bandwidth. We refer to this part of the connection establishment as \textit{MD associates with HN}. Similarly, we use the term \textit{MD disassociates from HN} if a mobile device drops the radio link established with its currently serving NAP.\footnote{In some wireless technologies, MD informs its currently serving NAP before it drops the radio link in order to allow for immediate reallocation of the wireless resources reserved for MD. Consequently, depending on the wireless technology in question, the disassociation process may involve more than dropping the radio link.}

Figure 1.10 illustrates the operational security-related phases of a connection establishment between a mobile device and its home network following a successful association. MD and the AS first negotiate the security suite to use. They then execute the authentication
protocol $a$ and the key-agreement protocol $ka$. During key agreement, both sides agree upon a master session key $K$. The master session key is then transferred to the EIPE. The EIPE and MD use the dynamic or static key-establishment process $ke$ to derive the encryption key $EK$ and the integrity-protection key $IK$ from $K$. In case of a dynamic key establishment process, $ke$ is a protocol with MD and EIPE as participants. In case of a static key establishment, no messages are exchanged between MD and EIPE during $ke$. This is illustrated by the dashed arrow for $ke$ in Figure 1.10. As soon as $EK$ and $IK$ are established, they are used to encrypt and integrity-protect the (data and/or control) traffic between MD and EIPE.

![Figure 1.10: Connection Establishment (See Page 1 for Notations Used in Figures)](image)

### 1.3 Security-Mechanism Negotiation

The security architecture of current wireless technologies (and current security protocols like IPsec, SSL, etc. in general) typically allow for the negotiation of one of several different security suites, leaving the final choice of the security suite to the users and network providers. Offering several alternative security suites has many advantages. First, it makes the technology break-through resilient: if an efficient attack against one of the security mechanisms in a security suite is found, this security suite can be avoided. The only change required to protect against newly found attacks is setting new policies. Second, new security mechanisms can easily be defined and integrated in the original architecture. This facilitates the upgrade of equipment and allows for the easy replacement of old mechanisms. Last but not least, as cryptographical strength typically trades with efficiency, alternative security suites can represent different points on an efficiency-security trade-off curve. Thus
offering different alternatives allows users and providers to choose how much efficiency they are willing to trade for specific cryptographical strengths.

Determining the cryptographical strength of a security mechanism is not an easy task. Some criteria to determine the strength of, for example, an encryption mechanism, are the key length, the amount of time thus far invested by researchers in analyzing the cipher, excluding certain well-known attacks and so on. Although some security mechanisms can be ordered according to their cryptographical strength in a way that will be accepted by any cryptographer, it is usually impossible to determine such a globally accepted “security level” for a security mechanism. Consequently, differences in the protection level of security mechanisms in fact have to be assumed to be subjective rather than objective. A user’s or provider’s security suite policy not only expresses which point on an efficiency-security trade-off curve he prefers, but also his subjective estimation of the cryptographical strength of the mechanism itself.

To select a security suite to use, the network and mobile device have to negotiate with each other. Various negotiation methods can be used. The negotiation can, for example, be completely predetermined, i.e., fixed during the pre-registration process, and lead to the selection of the same security suite on every connection establishment. It can also be dynamic, such that changes in the policy of the user or the network provider can be anticipated. The negotiation method can or cannot respect the network provider’s preference order on his policy $SS_{HN−allow}$ and the user’s preferences on the security suites $SS_{MD−allow}$ he allows to be used.

In the following sections, we describe five negotiation mechanisms (Method 1 to Method 5), discuss their advantages and disadvantages, and put them into the context of currently used and recently suggested negotiation methods. Method 1 to Method 3 describe state-of-the-art solutions. Method 4 adapts a mechanism suggested by Wang et al. in [179] for policy reconciliation in general to negotiating a security suite on connection establishment in a wireless access network. Finally, with Method 5 we present a new negotiation method, that takes the security policies as well as the preferences of MD and HN equally into account. We discuss in how far Method 5 is resistant against manipulation by the negotiating parties.

1.3.1 Related Work

The problem of security-mechanism negotiation on connection establishment in a wireless access network is a special sub-case of the general problem of security-policy reconciliation (sometimes also referred to as security-policy composition) which is an extensive field of research in itself. This field is concerned with three major problems: The first problem is specifying policy languages that is specifying a common syntax to represent policy information. The second one is specifying semantics for a policy language, such that policy statements are interpreted in the same way by all entities using the policy language. Finally, the third problem is composing or reconciling security policies.

Early work on security-policy languages (e.g., KeyNote [35], SPKI [58, 59], Binder [51] or SD3 [99]) focuses on access policy compliance testing for access requests rather than
policy reconciliation [113]. Recent work on policy reconciliation includes Wang et al.’s policy reconciliation with the help of a graphical tree representation of policies [179], the Dynamic Cryptographic Context Management (DCCM) [25, 26, 54] that extends the IPsec Security Policy System (SPS) described in [183], as well as the Ismene project [113]. While these works detail policy representations as well as semantics of special policy languages, we concentrate here on the negotiation only and express policies as a set of multi-dimensional vectors as in [54] or [183]. This is also the common practice in current wireless access technology standards.

Of the above mentioned work on policy reconciliation, only [179] considers preference orders on policy rules. The authors assume these preference orders to be partial and define the composition of two partial orders to a composite partial order in several different ways. The reconciliation mechanism they suggest then takes two policies and preference orders as input, computes the composite policy as a set of all policy rules that are compliant to both input policies, computes the composite preference order, and outputs one of the most preferred security suites according to the composite preference order. We describe how this approach can be used to negotiate a security suite between MD and HN in Method 4.

A deficit of this policy reconciliation mechanism, as well as of the others that do not respect preferences at all, is that it assumes the negotiating parties to be honest. As long as no preferences are considered, this assumption is reasonable. Yet as soon as preferences are taken into account, each of the negotiating parties can try to manipulate the negotiation to his advantage. Thus, if one party sends its complete policy along with its preferences to the other party, then the second party can enforce his preferences on the first one. Exchanging the security policies and doing the reconciliation on both sides does not solve the problem, as one party will always receive the policy of the other party before the other one receives his. Each party can therefore wait until it receives the policy of the other party and adapt its own policy to the received one before sending it to the other party.\footnote{It is important to note that during the security-mechanism negotiation, a mobile device and a network do not share any session-key material yet and have not authenticated each other. Thus, fair negotiation mechanisms that use encrypted commitments cannot be used for this purpose straightforwardly.}

The new negotiation mechanism Method 5 presented below partly solves this negotiation problem at the expense of several round-trip message exchanges between the negotiating parties. In each step of the negotiation protocol, each party only reveals one security suite it allows for. Method 5 makes the policy reconciliation in [179] less vulnerable to manipulation and yet always finds a pareto-optimal security suite whenever a common suite exists.\footnote{A security suite $ss$ is pareto-optimal with respect to the preference orders of MD and HN if and only if there is no security suite $ss^*$ HN and MD both prefer over $ss$.}

### 1.3.2 Negotiation without Preferences

The negotiation methods described in this section ignore the existence of any preference orders on allowed mechanisms and assume that HN and MD only either allow or disallow a security suite.

**Method 1.** During the registration process MD makes its complete policy $SS_{MD\text{-allow}}$
known to HN and vice-versa. Both parties agree upon one of the security suites they both allow to be used and create a database entry for the other party containing this security suite. On connection establishment, HN looks up the database entry for MD and reads the stored security suite. MD looks up its entry for HN and HN and MD subsequently use this security suite as described in Section 1.2.4.

The advantage of this simple method is its efficiency, as no negotiation messages have to be exchanged between the two parties during the connection establishment. Yet this method is completely inflexible. It does not allow either of the parties to change their policies without informing the other party. In particular, neither party can upgrade their equipment to support new security mechanisms and use them without prior off-line announcement. Similarly, neither party can easily react to newly found security holes in mechanisms they previously allowed without having to contact the other party first.

Method 2. Some of the missing flexibility of Method 1 can be gained by storing more than one alternative security suite in the corresponding database. HN then sends a choice of these alternative suites to MD, which chooses from the security suites offered by the network and sends back its choice to HN. In this negotiation method, HN can delete certain security suites from the database if it no longer supports them. Similarly, MD can avoid choosing certain mechanisms it previously supported during the time of registration, but no longer supports, by not choosing them if offered by HN. In this case, the storage of the security suites allowed for by MD during the time of registration ensures that the home network only offers allowable security suites to the mobile device. This reduces the amount of data exchanged during the negotiation. Nevertheless, fixing the set of possible choices makes it impossible for MD and HN to upgrade their policies to include new security suites without changing the database entries on the network side.

Method 3. To additionally gain this second type of flexibility, HN and MD do not store any information about the other’s policies, but they negotiate from scratch during the connection establishment. In this case, the home network sends its current $SS_{HN}^{allow}$ to MD during connection establishment, and MD chooses a mechanism from the intersection of its current $SS_{MD}^{allow}$ and sends its choice back to HN. To free MD from having to determine a security suite both allow, this method can be used in the other direction just as well. MD sends its current policy $SS_{MD}^{allow}$ to HN and HN determines a security suite they both allow and sends its choice back to MD. This is the way security mechanisms are negotiated in current wireless access technologies like GSM, UMTS, CDMA2000, IEEE 802.11i, as well as some security protocols like TLS or SSH.

As long as no preferences are considered, it is irrelevant whether MD or HN picks the security suite to use from the set of mechanisms they both support, as all allowed security suites are assumed to be equally well-accepted by the two parties. If the preferences of users and network providers are respected, which of the security suites MD and HN support is negotiated becomes important. This raises the question of who should finally decide upon which mechanism to use and how he should do so. Moreover, as soon as different preferences and security levels are considered, the question of whether a third person can “bid down” the negotiation between MD and HN arises. By interfering with the negotiation messages,
a third person could make HN and MD choose the lowest-level security suite they have in common. This problem is addressed in Section 1.3.4, while the question of who should control the negotiation and how he should do so is discussed in the next section. Note that a bidding-down attack is not possible if the policies used are pre-stored, as in Method 1 above.

1.3.3 Negotiation with Preferences

In this section, we assume that users and network providers have preference orders on their security policies $SS_{MD-allow}$ and $SS_{HN-allow}$. Providers and users are eager to prevail their preferences.

Implementing a negotiation mechanism that respects the preferences of the negotiating parties is particularly hard, as there are many possible different notions of a good choice. One intuitive notion of a desirable choice of a security suite is pareto-optimality. A security suite is **pareto-optimal** if there is no other security suite both the user and the home provider allow and which both would prefer. But, is a negotiation method that finds a pareto-optimal security suite really desirable? Assume the user is very careless about the policies he sets and more or less randomly chooses a security suite that is cryptographically quite weak, while the network provider, for example, a company, is very restrictive but has this particular security suite at the lowest end of its security suite policy. Then the user’s mobile device and the network will negotiate this low-level suite, although the user would not mind setting his policies in another way. This example demonstrates that the notion of what is desirable, as well as the symmetry of a negotiation mechanism, has to be carefully chosen to fit the area of application.

A negotiation method requires specifying how the preference orders of the negotiating parties are to be composed to a composite (partial) preference order and defining a negotiation mechanism that maximizes the composite partial preference order. A negotiation method is called **communication efficient**, if there is no other negotiation mechanism that leads to the same output on the same input but needs fewer messages to be exchanged between the negotiating parties.

As long as the negotiating parties are assumed to act honestly and respect the preferences of the other party during negotiation, efficiency is easy to achieve, as one party can simply send its policy to the other party. The second party then matches the policies according to some predefined agreed-upon negotiation rules and sends the result back to the first party. However, if the negotiating parties are greedy and respect their own preferences only, revealing the complete policy to the other party at once opens up the negotiation to adaptive behavior of the other party. Thus smaller commitments, possibly even single security suites, have to be exchanged in the security-mechanism-negotiation messages such that the parties reveal their preferences step by step and no one party obtains all the information on the security policy of the other party at once.

**Method 4.** Asymmetric negotiation mechanisms implement a biased negotiation and thus assume the preference order of one of the negotiating parties, without loss of generality,
party $A$, outweighs that of the other party $B$. Asymmetric negotiation is suitable for application scenarios in which one of the participants takes a higher financial or personal risk than the other as, for example, in the above-sketched scenario.

**Definition 1.3.1 Asymmetric Composition:** Let $\leq_A$ be a preference order of $A$ on $SS_{A\text{-allow}}$ and $\leq_B$ be the preference order of $B$ on $SS_{B\text{-allow}}$. Then the asymmetric composition (biased in favor of $A$) of the two preference orders to the composite preference order $\leq_{asym}$ on $SS_{A\text{-allow}} \cap SS_{B\text{-allow}}$ is in the asymmetric case defined as: for all $ss, ss^* \in SS_{A\text{-allow}} \cap SS_{B\text{-allow}}$,

$$ss \leq_{asym} ss^* \text{ iff } (ss <_A ss^*) \lor (ss =_A ss^* \land ss \leq_B ss^*).$$

This definition of optimality reflects that a security suite that is allowed by $A$ and $B$ and that is as highly valued by $A$ as possible should be chosen. In case there are several security suites supported by both, and $A$ prefers them equally well, one of the mechanisms $B$ prefers most amongst them is chosen. It is important to note that the composite preference order resulting from asymmetric composition is a total order.

A negotiation mechanism to implement this composite preference order can be defined as follows. $B$ sends its security-policy expression $SS_{B\text{-allow}}$ ordered according to its preference order to $A$. $A$ determines the elements $ss_{\text{max}}$ of $\max_{\leq_B} \max_{\leq_A} \{SS_{A\text{-allow}} \cap SS_{B\text{-allow}}\}$, chooses an arbitrary one, and informs $B$ of its choice. Obviously, $B$ cannot manipulate this negotiation mechanism without restricting $SS_{B\text{-allow}}$ and thus risking a failed negotiation.\(^{18}\) A dynamic negotiation mechanism requires at least two message exchanges such that Method 4 is communication efficient. For an efficient algorithm to compute an optimal security suite for the above-defined asymmetric composition, see [179].

**Method 5.** Let $A$ and $B$ be the negotiating parties, $k \in \mathbb{N}$ be the number of security suites in $A$’s security-policy expression $SS_{A\text{-allow}}$, and $l \in \mathbb{N}$ be the number of security suites in $B$’s security-policy expression $SS_{B\text{-allow}}$. Let $ss_{A_1} \leq_A ss_{A_2}, \ldots, \leq_A ss_{A_k}$ and $ss_{B_1} \leq_B ss_{B_2}, \ldots, \leq_B ss_{B_l}$ be $A$’s and $B$’s security policies ordered and numbered according to their preference orders.

**Definition 1.3.2 Pareto-Optimal Composition:** Let $\leq_A$ be the preference order of $A$ on $SS_{A\text{-allow}}$ and $\leq_B$ be the preference order of $B$ on $SS_{B\text{-allow}}$. Then we define the pareto-optimal composition of the two preference orders on the intersection $SS_{A\text{-allow}} \cap SS_{B\text{-allow}}$ as follows: for all $ss, ss^* \in SS_{A\text{-allow}} \cap SS_{B\text{-allow}}$,

$$ss \leq_{par} ss^* \text{ iff } ss \leq_A ss^* \text{ and } ss \leq_B ss^*.$$

\(^{18}\)It is important to note that we do not take learning effects of subsequent connection establishments into account here. In fact, any negotiation mechanism seems to be manipulatable over time.
With this composition method, a security suite is pareto-optimal if and only if it is maximal under the composite preference order. It is important to note that $\leq_{\text{par}}$ is only a partial order on $SS_A\text{--}allow \cap SS_B\text{--}allow$.

**Definition 1.3.3** A partial order on a set $S$ is a relation $\leq$ on $S$ with the properties

1. (Antisymmetry) If $s_1, s_2 \in S$ and $s_1 \leq s_2$ and $s_2 \leq s_1$ then $s_1 = s_2$
2. (Reflexivity) If $s \in S$ then $s \leq s$
3. (Transitivity) If $s_1, s_2, s_3 \in S$ and $s_1 \leq s_2$ and $s_2 \leq s_3$, then $s_1 \leq s_3$.

The following negotiation mechanism implements the pareto-optimal composite preference order and outputs a pareto-optimal security suite whenever a common security suite exists. If there is no such suite, the negotiation mechanism exits with a failure message.\(^{19}\)

We define $SS_A := \{ss_{A1}, \ldots, ss_{Ak}\}$, $SS_A^i := \{ss_{A_{i-1}}, \ldots, ss_{Ak}\}$, $SS_B := \{ss_{B1}, \ldots, ss_{Bk}\}$ and $SS_B^i := \{ss_{B_{i-1}}, \ldots, ss_{Bk}\}$. Furthermore, we initialize the boolean variables $m_A, m_B, last_A$, and $last_B$ with 0. During the negotiation, $A$ and $B$ exchange messages containing the next security suite ($ss_A$ or $ss_B$) of their ordered policy starting with their most preferred one, the boolean $m_A (m_B)$ indicating whether a match occurred on $A$’s side ($B$’s side) and a boolean $last_A (last_B)$ by which they indicated whether the current security suite is the last in $A$’s policy ($B$’s policy) or not. Upon receipt of the next security suite from $A$, $B$ checks whether the received $ss_A$ is in $SS_B$. Similarly, upon receipt of $ss_B$ from $B$, $A$ checks whether $ss_B \in SS_A$.\(^{20}\) Figure 1.11 illustrates the message flow for Method 5. $A$ and $B$ send these messages back and forth between them, until one of them finds a match and the other party confirms this match with a match confirmation message, or until one of them has committed to the least preferred security suite in his policy and no match was found (indicated by a failure message).

How $A$ determines whether or not a match occurred in step $i$ ($P_A^i$) and which message to return to $B$ is illustrated in Figure 1.12. Upon receiving $(ss_B, m_B, last_B)$ from $B$, $A$ first checks whether $m_B = 1$, that is, whether $B$ indicates a match. If this is the case, $A$ sends a match confirmation message back to $B$. If this is not the case, $A$ checks whether $last_B = 1$, that is whether $B$ indicates that the current commitment is its last one. If this is the case, $A$ checks whether $ss_B \in SS_A$. If this is the case, $A$ sets $m_A = 1$. If $A$ does not find a match, it sends a failure message back to $B$. If $last_B = 0$, $A$ checks whether it reached the end of its own policy ($i = k$). If this is the case, $A$ sets $last_A = 1$. $A$ sets $ss_A := ss_{A_{i-1}}$. $A$ sends $(ss_A, m_A, last_B)$ back to $B$. Upon receiving $(ss_A, m_A, last_B)$, $B$ proceeds in exactly the same way, exchanging $A$ with $B$ in Figure 1.12.

---

\(^{19}\)The presented mechanism, in fact maximizes the sum of the ranks assigned by $A$ and $B$ to their security suites corresponding to their preferences.

\(^{20}\)In this negotiation method, $A$ compares $ss_{B1}$ with $ss_{A1}, \ldots, ss_{Ak}$ and $B$ compares $ss_{Ai}$ with $ss_{B1}, \ldots, ss_{B_{i-1}}$ such that only $A$ and not both parties compare the security suites with the same index. This is where the asymmetry in $SS_A$ and $SS_B$ comes from.
Example. Assume A has the ordered security suite policy $ss_1 \leq_A ss_2 \leq_A ss_3$ and B has the ordered security suite policy $ss_2 \leq_B ss_5 \leq_B ss_3 \leq_B ss_1$. In the first step, A sends $ss_3$ to B. B answers with $ss_1$. A next commits $ss_2$, B answers with $ss_3$ and $m_B = 1$, as $ss_3$ is one of the cipher suites A has already committed to allow. A confirms the reception of the match and A and B subsequently use $ss_3$. Obviously, $ss_3$ is a pareto-optimal security suite for the given preference orders of A and B. It is interesting to note that $ss_1$ in this situation is also a pareto-optimal solution, although intuitively $ss_3$ is a much better choice than $ss_1$. This shows that pareto-optimality as optimization metric could still lead to choices that intuitively would not be considered optimal.

Discussion. It is interesting to note that in the two-party case, either no common security suite exists or a common pareto-optimal security suite exists. Otherwise, for every common security suite $ss$, there would be a common security suite $ss^*_1$ that A and B prefer over $ss$. As $ss^*_1$ cannot be pareto-optimal, there is another security suite $ss^*_2$ that is preferred by A and B and so on. As there is only a finite number of security suites, this leads to a contradiction.

If the negotiating parties act according to the protocol, the above negotiation protocol always finds a pareto-optimal solution or exits with a failure message if there is no common security suite. This is due to the fact that A and B start their negotiation from their most preferred security suites. Assume $ss$ is the output of the negotiation protocol and the output is not pareto-optimal. Then, there is a $ss'$ that is supported by both A and B, with $ss \leq_A ss'$ and $ss <_B ss'$. In this case, B would have committed $ss'$ to A in an earlier step than $ss$ and thus $ss'$ would have been chosen as the output. Thus, if the negotiation protocol does not exit with a failure message, a pareto-optimal solution is found. On the
other hand, if the protocol exits with a failure message, then A and B obviously do not have a security suite in common.

As long as A and B know nothing about each other’s policies and preference orders, they gain nothing from changing their policies or the preference order on their policies on the current run of the negotiation mechanism.

On the first execution of the above negotiation mechanism between them, A and B learn part of each other’s security policy expression and the preference order on this part. If \( ss^* \) is the result of the negotiation mechanisms and A and B stick to the protocol, A learns all \( ss_B \) with \( ss_B \geq_B ss^* \) and B learns all \( ss_A \) with \( ss_A \geq_A ss^* \).

Assume that on a second run of the protocol, A tries to get B to negotiate a cipher suite \( ss_A \geq_A ss^* \) with B (this is the only realistic goal A can have). Assume, furthermore, that for this purpose, A deletes \( ss^* \) from its policies. As A knows nothing about the security

---

**Figure 1.12**: A’s Decision in Step \( i \) of Method 5 (\( P_A^i \) in Figure 1.11)
suites $ss_B \leq ss^*$, $A$ gains nothing from committing security suites other than the ones it prefers most to $B$. $B$ can observe the missing $ss^*$ in the negotiation and may either react in rejecting the negotiation all together or may react by deleting all $ss_{A_j} \geq_A ss^*$ from its policy ($B$ knows these from the first negotiation). Thus $B$ can ensure that $A$ gains nothing from deleting $ss^*$ from its policy.

Although this negotiation method is surprisingly resistant against some kind of manipulation, it cannot resist attacks that aim to learn only during the first runs of the negotiation (letting the negotiation fail) and then using the obtained knowledge to enforce the security suite to use: If $A$ changes its policy on the first run of the negotiation to a set of security suites he can be quite certain that $B$ does not support, then $A$ can make the negotiation fail. A failing negotiation means, that $A$ gains knowledge of $B$’s complete policy. $A$ can thus pick the maximal security suite of $B$’s policy under its own preference order and redefine its own policy to have this security suite in the first place and mechanisms $B$ does not support in all other places. $A$ and $B$ will then negotiate $A$’s best choice. This type of manipulation can partly be prevented by allowing security policies to change over time only by reordering or deletion.

### 1.3.4 Protecting Against Bidding Down

Unauthorized changes to the messages exchanged between the negotiating parties are a threat to any dynamic security-mechanism negotiation. An attacker that can, for example, change $SS_{MD-allow}$ sent from MD to HN in Method 4 could change $SS_{MD-allow}$ to include only the lowest level security suite MD allows for. HN would thus have to choose this lowest-level security suite or reject the connection altogether. Bidding down attacks like this are well known, for example, for GSM [89] and SSL v.2.0 [174]. One possible solution to this problem is integrity-protecting the negotiation messages. Unfortunately, at the time of negotiation, the parties have not yet agreed upon keys or mechanisms for integrity protection yet. In SSL, as well as UMTS, this problem is solved by replaying the received $SS_{MD-allow}$ at the end of the authentication and key agreement protocols in an integrity-protected message. Consequently, an attacker can only change $SS_{MD-allow}$ without MD’s notice if he can forge the integrity protection. This solution only works if the technology does not support any weak integrity-protection mechanisms and, in particular, if integrity protection is mandatory.
Chapter 2

Roaming

The goal of state-of-the-art inter-provider roaming procedures is to provide network access to users in a wider coverage area than the one offered by a single provider, with as little extra effort for the user as possible. For this purpose, network providers enter into roaming agreements with each other. Users are required to register with only one provider, their home provider. In the registration process, the user and his home provider exchange credentials and agree upon a roaming profile that includes a roaming region for the user. The credentials established with his home provider enable the user not only to access the network operated by its home provider, but also the ones operated by those roaming partners of its home provider in the user's roaming region.

Roaming support can be offered by commercial providers, e.g., two mobile phone operators or two WLAN Hotspot providers, or by non-commercial providers, e.g., the WLAN networks of two companies or two departments of a university. In the commercial case, a user and his home provider agree upon roaming charges in the registration process. The user receives only one bill from his home provider, which charges MD and reimburses FN for service provisioning. The home provider and the foreign provider agree upon charging issues in their roaming agreement.

A recent trend is to integrate more than one wireless interface into a single mobile device. This allows users to benefit from the advantages of different technologies, depending on which one is available at their current location. Similar to the inter-provider case, a user should be able to access different networks of different technologies and optionally different providers with only one subscription and, in the commercial case, only one bill.

The main security challenge in roaming across providers and technologies is that upon roaming, a foreign network and a mobile device have to authenticate each other and agree upon cryptographic keys for encryption and integrity protection without sharing any pre-established credentials.

This chapter includes two main contributions. The first one is a formal, technology-independent model of roaming procedures in general and the second one is the introduction of a new approach to public-key-based inter-provider roaming. In the following paragraphs we detail these two contributions.
We model, categorize, and discuss roaming authentication and key-agreement protocols for wireless access networks. For the modeling, we analyzed the most widely spread standardized roaming procedures, namely the inter-provider roaming procedures in the mobile phone networks GSM, CDMA2000, UMTS, and the roaming protocols designed for roaming across GSM-UMTS, GSM-WLAN, GSM-GPRS, UMTS-WLAN and UMTS-CDMA2000 [62, 9, 153, 88, 12]. Our modeling is furthermore based on recent suggestions to enhance the standardized roaming procedures (e.g., [85, 141]) and other suggestions for inter-provider roaming, (e.g., [29, 156, 44, 43, 189, 167, 65, 178, 24, 23, 19, 118, 81, 131, 98]).

None of the aforementioned work addresses the problem of security-mechanism negotiation during handover. We explicitly address this problem. We show that in some roaming scenarios it is in HN’s interest to influence the choice of security suite used between FN and a roaming MD. We introduce and discuss various negotiation methods that can be used to negotiate the security suite and that enforces policies of HN, FN and MD.

Most of the aforementioned work offers a security solution for roaming for a particular technology or between a particular pair of technologies. Instead, we take a technology-independent viewpoint here. Other more general work, like [29, 81, 156], is restricted to a certain credential type. We model and discuss roaming protocols based on all types of credentials. In particular, we classify roaming protocols according to the amount of control they leave to the home provider to accommodate changes in roaming agreements and roaming profiles, the amount of interaction required between MD, FN and HN, as well as the knowledge of confidential information gained by each party. Moreover, we defined the following design goals for security solutions upon roaming: an ideal roaming authentication and key-agreement protocol should minimize the authentication traffic required between a foreign network and the home provider as well as between a foreign network and a mobile device, it should allow for easy handling of changes in roaming profiles and agreements, and derive cryptographic keys in the foreign network, where they are used.

While some state-of-the-art roaming authentication protocols are specifically designed to minimize the amount of traffic required between FN and HN (e.g. [156]) none of them currently supports key derivation in the foreign network. We here present a new technology-independent approach for public-key-based inter-provider roaming that addresses this shortcoming of other solutions. In our approach, secret-sharing techniques are used to facilitate inter-provider roaming within or across different technologies. In particular, we show that this technique potentially minimizes the number of messages required to be exchanged between a foreign provider and the home provider, allows for the derivation of cryptographic keys in the foreign network, and allows for the easy handling of changes in roaming profiles and agreements by engaging the home provider in every roaming instance [121].

In addition to the two main contributions detailed above, we briefly discuss some issues that are related to inter-provider roaming but considered largely out of scope of this work. One these issues is intra-provider roaming, the second one is accounting, and the third one is the protection of the confidentiality of a mobile device’s identity upon roaming.

Finally, we discuss a future direction of research, namely roaming mediators. State-of-the-art roaming procedures are based on pairwise roaming agreements between network
providers, which results in an overall number of roaming agreements that is quadratic in the number of network providers. The introduction of roaming mediators to the scene reduces the number of roaming agreements, as each provider has a single agreement with a roaming mediator. The roaming mediators, in turn, may have pairwise roaming agreements with each other. Roaming agreements between two providers are then setup with help of the roaming mediators on the fly.

Outline. In Section 2.1, we first describe the relationships between a user and its home provider, as well as between a home provider and its roaming partners. We then define roaming authentication and key-agreement protocols, discuss our assumptions on roaming security policies, and describe different roaming security-mechanism negotiation methods. We proceed with a classification of current roaming protocols. In Section 2.2, we present our key-splitting approach for inter-provider roaming. Roaming across different wireless access technologies is discussed in Section 2.3. Finally, we briefly summarize some related research issues (Section 2.4) and complete this chapter with an extensive related-work section (Section 2.5).

2.1 Inter-Provider Roaming - Modeling and Classification

2.1.1 Roaming Agreements and Registering for Roaming

In state-of-the art roaming, a network provider has a roaming agreement with other network providers, its roaming partners. In such a roaming agreement, two providers agree upon which services roaming users should be able to use\(^1\) and optionally agree upon the authentication and key agreement procedures, as well as the cipher suites to be used. In the commercial case, the roaming agreement additionally fixes charging and billing issues, as well as legal terms between the providers. A roaming agreement is typically valid with respect to two particular networks, each operated by one of the providers. When entering a pairwise roaming agreement, two network providers establish a trust relationship and exchange credentials.

On the user’s side, a user and his home provider fix the user’s roaming profile during registration. The roaming profile includes a set of foreign networks as well as the services a user will be able to use upon roaming. During the registration, the user and its home provider may additionally exchange information on their roaming security policies and store these on the user’s MD and in SC. In the commercial case, a user and his home provider also fix the roaming charges during the registration.

2.1.2 Security Mechanisms

One of the security challenges upon roaming is that FN and MD have to mutually authenticate each other without any prior established trust relationship. Consequently, the mutual

\(^1\)E.g., two mobile phone network operators may enter a roaming agreement with regard to voice services but not with regard to GPRS service.
authentication between FN and MD differs from the mutual authentication between a mobile device and its HN modeled and defined in the last chapter (see Section 1.2.2.2) in an important way: MD and HN authenticate each other with respect to individual identities. The credentials HN and MD exchange during registration allow them to uniquely identify each other and then verify the binding between their identities and the credentials, thus authenticating each other. As opposed to this, upon roaming, MD and FN are not assured of each other’s identity, but rather of HN’s authorization of the roaming instance. Upon roaming, FN has to prove to MD that it is authorized by HN to offer service to MD. MD has to prove to FN that it is authorized by HN to use FN’s services. In case of commercial network providers, HN’s authorization of the roaming instance assures FN that HN is willing to reimburse FN for service provisioning. A roaming authentication protocol can be defined in the following way:

**Definition 2.1.1** A mutual roaming authentication \((ra)\) between MD and \(AS_{FN}\) is a protocol between MD, \(AS_{FN}\), and optionally \(AS_{HN}\), whereby MD and \(AS_{FN}\) are assured of the home provider’s authorization of the roaming instance based on a particular set of credentials established between MD and \(AS_{HN}\) and optionally also based on \(AS_{FN}\)’s prior interaction with \(AS_{HN}\).

Aside from the authentication itself, \(AS_{FN}\) and MD have to agree upon a master session key upon roaming, such that they can derive encryption and integrity protection keys in order to protect data and control traffic between MD and EIPE\(_{FN}\). The roaming key-agreement may additionally involve \(AS_{HN}\).

**Definition 2.1.2** A roaming key-agreement protocol \((rka)\) between MD and \(AS_{FN}\) is a protocol between MD, \(AS_{FN}\), and optionally \(AS_{HN}\), whereby a shared master key \(K\) is established between MD and \(AS_{FN}\). This key is derived as a function of information ideally contributed by each party such that no party can predetermine the resulting value.

After a successful key-agreement, \(AS_{FN}\) uses a key-transfer mechanism \(kt\) (see Def. 1.2.8) to transfer the master key \(K\) to EIPE\(_{FN}\).

MD and EIPE\(_{FN}\) use a key-establishment process \(ke\) (see Def. 1.2.9) to derive the data-protection keys \((EK, IK)\) from the master key \(K\). MD and EIPE\(_{FN}\) subsequently use these keys as input to an encryption mechanism \(em\) (Def. 1.2.4) or an integrity-protection mechanism \(im\) (Def. 1.2.7) to protect their MAC layer connection.

During connection establishment upon roaming, MD and FN have to negotiate the roaming authentication protocol \(ra\), the roaming key-agreement protocol \(rka\), and the cipher suite \((ke, em, im)\) to use. This security-suite negotiation may require HN’s engagement.

\(^2\)We speak of a roaming instance whenever MD tries access a foreign network, i.e., whenever MD roams to FN.
2.1.3 Security Policies

Throughout this work we assume that for each wireless access technology, there are technology specific sets RA and RKA of roaming authentication protocols \( ra \) and roaming key-agreement protocols \( rka \). Each roaming authentication protocol is based on a certain credential type. Two roaming authentication protocols designed for the same technology may be based on different credential types.

In addition, we assume that there is a technology-specific set of roaming security suites \( RSS \) that is a subset of the Cartesian product of RA, RKA, and the technology-specific sets of key-establishment, encryption and integrity-protection mechanisms, KE, EM and IM (see Section 1.2.3):

\[
RSS \subseteq RA \times RKA \times KE \times EM \times IM.
\]

The elements \( rss \) of \( RSS \) are referred to as roaming security suites here. The set \( RSS \) is the set of all technologically possible and standard-wise allowed combinations of security mechanisms for the given technology. We assume here that this set is known to every network provider and every user.

MD and each network have policies with respect to the roaming security suites they allow to be used. If MD requests access to its HN, only HN’s and MD’s policies have to be taken into account. On roaming to FN, current solutions only take policies of FN and MD into account. This may, however, not be sufficient. As HN’s authorization of the roaming instance is assured to MD and FN, HN may be held responsible for attacks against the security goals of MD or FN. In particular, for example, in the commercial case, HN may have to reimburse FN for service provisioning even if the roaming authentication between MD and FN was broken and an unauthorized party got access to FN on behalf of MD. Nevertheless, whether or not HN is held responsible for attacks like this depends on the roaming agreement between HN and FN, in particular on the legal terms HN and FN agreed upon. On the other hand, HN may give its pre-registered users guarantees regarding their level of protection upon roaming to FN. For example, HN may give MD guarantees on a secure roaming authentication or on a certain level of MAC layer protection.

We therefore assume that not only MD and FN have roaming policies with respect to the security suites to be used upon roaming, but also HN. MD, HN and FN express these policies by pre-defining subsets of \( RSS \). We denote these policy expressions by \( RSS_{MD-allow} \), \( RSS_{HN-allow} \), and \( RSS_{FN-allow} \). They specify which security suites MD, FN, and HN allow to be used if MD roams to FN.\(^3\) Optionally, MD, HN, and FN may have a preference order \( \preceq_{MD} \), \( \preceq_{FN} \) or \( \preceq_{HN} \) on their roaming policies.

In the remainder of this work, we use the notation \((R)SS = SS \cup RSS\) to denote the elements of \((R)SS\) with \((r)ss\). Similarly, we denote the elements of \( A \cup RA \) with \((r)a\)

\(^3\)To be precise, we assume that MD has the same roaming security policy for each foreign network of the same technology. Similarly, the home provider has the same roaming security policy for each foreign network that supports the same technology.
and the elements of $KA \cup RKA$ with $(r)ka$. Furthermore we denote the policy expression $RSS_X^{allow} \cup SS_X^{allow}$ with $(R)SS_X^{allow}$.

### 2.1.4 Security-Mechanism Negotiation

Upon roaming, MD and FN have to negotiate a security suite they both allow. In state-of-the-art roaming procedures MD’s and FN’s policies are typically taken into account, while HN’s policy is not enforced. MD and FN then negotiate a security suite as described in Method 6.

**Method 6:** MD and FN negotiate a security suite by replacing HN with FN, $SS_{MD}^{allow}$ with $RSS_{MD}^{allow}$, and $SS_{HN}^{allow}$ with $RSS_{FN}^{allow}$ in Method 3 described in Section 1.3.

Method 6 does not take any preferences of MD or FN on the security suites they allow to be used into account. As detailed in Section 1.3 this contradicts the observation that user’s and providers typically deem different security suites to offer different security levels. If MD and FN additionally want to take their respective preferences into account, they can use Method 4 or Method 5 with the same replacements mentioned above (see Method 6).

The fact that current roaming procedures do not enforce HN’s policy with respect to the security mechanisms allowed to be used upon roaming becomes a threat to HN as soon as HN may is held responsible, e.g., for service theft attacks against FN by means of certain terms in the roaming agreement. In this case it is crucial for HN to enforce its policy upon with respect to the security mechanisms used upon roaming.

If HN’s policies are to be respected upon roaming, either HN can be engaged in the negotiation protocol itself or certain policies may be fixed in the roaming agreement and as part of the roaming plan for MD. This leads to a trade-off between fast negotiation and HN’s flexibility in setting policies. In the following we briefly describe three new security-mechanism negotiation methods that respect HN’s policies. The first one (Method 7) does not take any preferences of MD, FN or HN into account. Method 8 takes preferences of MD, FN, and HN into account. In both of these methods, HN is not engaged into the negotiation online. HN reveals its policy (and in case of Method 8 also its preferences) to MD and FN. In Method 8 HN is engaged into the negotiation online and can is therefore more flexible with respect to changing its policy. In all of the three methods HN has to trust at least FN or MD to respect its policy. However, FN (MD) alone cannot manipulate the negotiation result in a way that it does not comply to HN’s policy without MD (FN) detecting this manipulation.

**Method 7:** If no preferences are to be respected and HN is not to be engaged in the negotiation online, HN can reveal its policy expression $RSS_{HN}^{allow}$ to MD in the pre-registration and to FN in the roaming agreement. Both MD and FN then use the intersection of their own policy expressions with HN’s, rather than solely their own expressions. MD and FN can then use Method 3 of Section 1.3 to negotiate a security suite. Note that in this case, HN has to trust MD and FN to respect its policy. Nevertheless, revealing its policy to both parties ensures that both parties would have to ignore HN’s policies in order to choose a mechanism HN does not allow to be used.
Method 8: If the preferences of MD, HN, and FN are to be taken into account and HN is not to be engaged in the negotiation online, then MD and HN (HN and FN) can reconcile their policies during registration (on entering a roaming agreement). The policy reconciliation mechanism respects their preferences with the desired weights and helps the two parties derive a total preference order on the intersection of their policies. An example for a composition of two total preference orders that outputs a total composite preference order is the asymmetric composition defined in Definition 1.3.1.\footnote{The pareto-optimal composition (see Definition 1.3.2) cannot be used here, as it outputs only a partial order.} MD can then store the reconciled policy expression together with the combined preference order. The same holds for FN. Upon roaming, MD and FN use the stored reconciled policy expressions and composed total preference orders to negotiate the security suite to use, for example, with the help of Method 4 or Method 5 of Section 1.3.

Method 9: In order to gain more flexibility for HN with respect to changing policies, HN can be engaged online in the negotiation. This is particularly easy to achieve if HN is required to be online during the roaming authentication and key-agreement protocols anyway. How the security-suite negotiation between HN, MD, and FN can efficiently be integrated into the connection establishment largely depends on how HN is engaged in the authentication and can only be further discussed for each particular roaming protocol. To give just one example, MD could send FN its roaming policy expression $RSS_{FN-allow}$ in a first step. FN could then compute the intersection of MD’s and its own $RSS_{MD-allow}$ and forward the intersection together with a request for authentication information to HN. HN could then pick one of the suites in the received intersection and command FN and MD to use this security suite. In order to prevent FN from changing the selected security suite, HN has to add an authentication token to the selected suite, that can be verified by MD.

2.1.5 Classification of Roaming Authentication and Key Agreement Protocols

The authentication protocol, the key agreement, and the security-mechanism negotiation on roaming can be implemented in many different ways. In this section, we detail some distinguishing properties of roaming authentication and key-agreement protocols, discuss their advantages and disadvantages, determine the possible property combinations, and classify existing roaming solutions according to their properties.

Public-Key-Based versus Non-Public-Key-Based. Roaming authentication and key-agreement protocols can be based on public-key certificates or can be non-public-key-based. In general, public-key-based authentication protocols have the advantage of allowing two parties to mutually authenticate each other without requiring any prior trust relationship between them. As such, public-key-based methods seem to be the method of choice for authentication upon roaming. However, they raise several difficulties.
For once, many current wireless access technologies do not support public-key certificates. Furthermore, if MD and AS\textsubscript{FN} authenticate each other based on individual certificates, both have to validate these certificates during the authentication. The validation includes verifying CA’s signature on the certificates, and checking their revocation status. Even if we assume that MD’s and FN’s certificate have been signed by a CA both trust directly (which enables them to easily verify CA’s signature), MD and AS\textsubscript{FN} still have to check the revocation status of each other’s certificates. This is particularly difficult for MD, as MD has to check FN’s certificate in order to get network access but needs network access in order to access remote resources for status checking. If MD does not trust the CA that issued FN’s certificate directly, this task becomes even more difficult: MD has to obtain a chain of certificates with a trusted certificate as root and validate each certificate in this chain in order to validate FN’s certificate. Recently it has been suggested (e.g., [29]) that upon roaming, MD should delegate the validation of FN’s certificate to a trusted third party. In this case, MD has to be sure of the revocation status of the certificate of the trusted third party only. However, this method causes additional authentication traffic between MD and the trusted third party (forwarded by FN), and thus increases the overall load on the backbone network, potentially delaying the authentication. Additionally, this solution does not address a second problem arising from the use of individual certificates for MD and FN.

All relevant information regarding the roaming profile (e.g., the roaming region) of a MD has to be encapsulated in MD’s certificate, for example, in the form of attributes. Consequently, if the roaming profile of MD changes, a new certificate has to be issued for MD. Similarly, all relevant information on the roaming agreement between FN and HN has to be encapsulated in FN’s certificate. Using public-key certificates without online engagement of HN in the authentication on roaming can thus not easily accommodate changes in roaming plans and roaming agreements.

To avoid the above shortcomings some public-key-based authentication protocols that engage HN online have been suggested. In the most extreme case (as, e.g., in [189]) AS\textsubscript{HN} and MD mutually authenticate each other based on individual certificates and agree upon a master key, while FN only forwards the authentication traffic between them. HN then uses the credentials pre-established with FN to securely transfer this master key to FN. MD and FN mutually assure each other of their authorization indirectly by proving possession of the master key to each other. In this case, MD has to validate HN’s certificate only. Moreover, the information on a roaming plan and a roaming agreement is available and under full and timely control of HN. In other cases of roaming authentication (e.g., [121]), HN engages in the authentication in a different way and rather assists FN and MD in authenticating each other, than authenticating both parties itself.

**Online/Offline Engagement of HN.** State-of-the-art non-public-key-based authentication and key-agreement protocols as well as some public-key-based ones require HN’s engagement in the roaming authentication (e.g [156, 62, 9]). HN’s interaction can either be offline before the actual authentication or online during the actual authentication. In
the offline case, HN typically provides FN with some security-related information with the help of which MD and FN can authenticate each other. Offline engagement of HN generally trades an efficient authentication protocol (no round-trip message exchanges with HN during authentication) against the freshness of HN’s authorization of the roaming instance. In some roaming protocols, the first of a certain number of authentications between FN and MD requires HN’s online engagement while the subsequent ones involve only FN and MD. A method like this tries to minimize round-trip message exchanges to HN and yet guarantee the freshness of HN’s authorization.

Note that some public-key-based protocols designed for roaming use public-key-based authentication to authenticate FN to MD, yet use a non-public-key-based method to authenticate MD to FN.

**Key Derivation by HN or FN.** In any public-key-based or non-public-key-based authentication and key-agreement protocols, MD and FN have to establish a master key $K$. From $K$, they subsequently derive the integrity and encryption key to protect their connection. If AS$_{HN}$ is involved in the roaming authentication, the master key $K$ can be derived by HN or by FN. If HN derives $K$, HN may derive $K$ offline before the actual roaming authentication or online, during the authentication. If HN derives $K$ (offline or online), HN has to establish a secure channel to FN in order to transfer $K$ to FN. In case FN derives $K$, HN may or may not be able to derive $K$ itself as well and thus may or may not get knowledge of $K$. As only FN and MD are required to have knowledge of $K$, it is desirable for a roaming protocol to derive $K$ in FN and not in HN. Note that even if FN derives $K$, HN and FN may still require a secure channel between them, as HN may have to transfer secret information contributing to the key to FN during or before the authentication.

**Design Goals for Roaming Authentication.** From the above discussion, we derive the following general design goals for roaming authentication and key-agreement protocols. Ideally, roaming authentication and key-agreement protocols should minimize the authentication traffic between AS$_{FN}$ and AS$_{HN}$. They should be able to easily and timely accommodate changes in the roaming profile of a mobile device as well as changes in roaming agreements between two providers. In particular, roaming agreements and roaming profiles should be easy to revoke and the revocation should be effective as isochronous as possible. The same holds true for new roaming agreements or new extensions of the roaming profiles of MD. The master key $K$ agreed upon between AS$_{FN}$ and MD should be derived by AS$_{FN}$ to avoid unnecessary key transfers.

In the following three subsections, we describe three types of roaming authentication and key-agreement that we distinguish according to their properties as summarized in Table 2.1.

---

5HN neither online nor offline involved.
### 2.1.5.1 Type 1 Roaming Procedures

In Type 1 roaming procedures MD and FN mutually authenticate each other based on individual public-key certificates. MD and FN are required to validate each other’s certificates during authentication. The information needed for certificate validation is provided by HN or a trusted third party before, during, or after the authentication. MD and FN negotiate the security suite to use without interacting with HN. HN can try to ensure its policies offline by revealing its policies to MD in the pre-registration process and to FN upon entering the roaming agreement.

The advantage of this procedure type is that FN generates the master key $K$ itself and HN does not get knowledge of $K$. In particular, no secure channel for key transfer from HN to FN is required. Another advantage of a method like this is that it does not require any traffic on the backbone network other than traffic related to the revocation status of certificates. On the other hand, as any public-key-based method that uses individual certificates for each FN, this method cannot easily handle changes in roaming plans or roaming agreements which makes it inflexible. In addition, as discussed earlier, it requires particularly adapted solutions to enable MD to validate the certificate presented by FN.

Figure 2.1 illustrates the security-mechanism negotiation as well as authentication and key agreement upon roaming to FN in this case. For simplicity, the possibly required certificate revocation status checks before, during, or after authentication are not illustrated.

The roaming authentication protocol $ra$ and the key-agreement protocol $rka$ are protocols between AS$_{FN}$ and MD. AS$_{FN}$ and MD authenticate each other based on individual public-key certificates. In the key-agreement protocol, the master key $K$ is generated in AS$_{FN}$ and MD. AS$_{FN}$ transfers $K$ to EIPE$_{FN}$ using a key-transfer mechanism $kt$. AS$_{HN}$ does not gain any knowledge on $K$ and no secure channel between AS$_{HN}$ and AS$_{FN}$ is required. AS$_{HN}$ is, if at all, only engaged in the authentication by providing information about the revocation status of certificates. For this purpose, AS$_{FN}$ and MD may contact AS$_{HN}$ before, during, or immediately after the authentication. AS$_{HN}$ may thus be online or offline or not at all engaged in each roaming authentication. See Table 2.1 for the properties of Type 1 procedures.

<table>
<thead>
<tr>
<th></th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>HN online</td>
<td>+/-/o^a</td>
<td>+</td>
<td>+/-</td>
</tr>
<tr>
<td>FN derives $K$</td>
<td>+</td>
<td>-</td>
<td>+/-</td>
</tr>
<tr>
<td>HN derives $K$</td>
<td>-</td>
<td>+</td>
<td>-/+</td>
</tr>
<tr>
<td>HN knows $K$</td>
<td>-</td>
<td>+</td>
<td>-/+/+</td>
</tr>
<tr>
<td>Sec. chan. req.</td>
<td>-</td>
<td>+</td>
<td>-/+</td>
</tr>
<tr>
<td>p-k-based</td>
<td>+</td>
<td>-</td>
<td>+/-</td>
</tr>
</tbody>
</table>

Table 2.1: Types of Roaming Procedures
2.1.5.2 Type 2 Roaming Procedures

A Type 2 roaming procedure can either be public-key-based or non-public-key-based. $AS_{FN}$ forwards all authentication and key-agreement protocol messages between MD and $AS_{HN}$. Consequently, $AS_{HN}$'s engagement in the roaming procedure is online. MD and $AS_{HN}$ authenticate each other based on their pre-established credentials. $AS_{HN}$ and MD agree upon a master key $K$ and $AS_{HN}$ transfers this master key to $AS_{FN}$ after successful authentication. $FN$ and MD subsequently assure each other of $HN$’s authorization by proving their knowledge of the master key $K$ to each other. $AS_{HN}$ and $AS_{FN}$ use the credentials established as part of their roaming agreement to establish a secure channel for the key transfer $kt^*$. The necessity of a secure channel is one of the disadvantages of this type of roaming protocol. Another disadvantage is that $AS_{FN}$ has to forward all authentication traffic between MD and $AS_{HN}$. This results in a major load on the backbone network connecting $AS_{HN}$ and $AS_{FN}$. The advantage of this method is that $HN$ can easily control each roaming instance and timely react to changes in the roaming profile of MD or changes in the roaming agreement with FN. Moreover, $HN$ can actively take part in the security-mechanism negotiation and allow or forbid the use of certain encryption and integrity-protection mechanisms after authentication. Figure 2.2 illustrates this type of roaming procedure.

---

6$FN$ does not have to support or even know the actual authentication protocol used between $AS_{HN}$ and MD.
The security-mechanism negotiation can be implemented in many different ways. We only give one example here: MD sends $RSS_{MD\text{-allow}}$ to HN. HN computes the intersection of $RSS_{HN\text{-allow}}$ with $RSS_{MD\text{-allow}}$ and sends the intersection (together with the result of the authentication) to FN. FN then selects one of the mechanisms in the intersection and acknowledges its choice to MD. Note that this is not sufficient to ensure that HN’s policies are in fact respected. In particular, FN could intercept $RSS_{MD\text{-allow}}$ and choose one of the mechanisms allowed by MD and itself, but not HN. If HN takes the financial or non-financial risk during roaming, HN will want to prevent FN from doing this. For example, HN could acknowledge $RSS_{HN\text{-allow}}$ to MD in an authenticated way using the pre-established credentials. As a consequence, HN still cannot be sure that its policy is respected, but FN and MD would have to cooperate in order to use a security suite HN does not allow.

### 2.1.5.3 Type 3 Roaming Procedures

While Type 1 and Type 2 procedures are the extremes, Type 3 covers all methods in between the first two. Type 3 roaming procedures require HN’s online or offline interaction for more than just providing information on the revocation status of certificates; rather Type 3 procedures split the load of the authentication between HN and FN. Type 3 procedures can be public-key-based or not and may require HN to be online or not. In some of them, HN will generate the master key and transfer it to FN, thus requiring a secure channel between HN and FN. In others FN will derive the master key itself, while requiring some secret input of HN to do so. In yet others, FN may be able to derive the master key without HN’s interaction. It is important to note that in most Type 3 procedures suggested so far, MD communicates with AS$_{FN}$ only. In some protocols suggested for authentication across different technologies in wireless overlay networks\footnote{A network is said to overlay another network if its cells overlay the coverage area of the other network completely.} (e.g., [45, 169]), MD is required to communicate with both AS$_{HN}$ and AS$_{FN}$ simultaneously. Figure 2.3 illustrates a Type 3 procedure.

For Type 3 roaming procedures, no general statements on when or how to negotiate security mechanisms can be made.

### 2.2 Enhancing Roaming Protocols by Means of Secret-Sharing

State-of-the-art public-key-based roaming protocols are either of Type 1 or are public-key-based Type 2 procedures. As discussed before, the shortcomings of the former approach are problems regarding certificate validation on the mobile station side as well as costly handling of roaming plan changes. The shortcomings of the latter approach include the requirement for a secure channel between AS$_{FN}$ and AS$_{HN}$, as well as a large amount of round-trip message exchanges between AS$_{FN}$ and AS$_{HN}$ during authentication. The key...
idea of using secret sharing in this setting is to address these shortcomings by replacing the individual certificates for FNs by means of a suitable key splitting between HN and FNs in combination with issuing a certificate for HN only. In more detail, this works as follows:

Every HN is issued one roaming certificate. Assuming HN has a pairwise roaming agreement with \( l \) foreign networks \( FN_1, \ldots, FN_l \), HN splits its secret roaming key \( R \) into \( l \) different pairs of shares \((R_{HN,i}, R_{FN,i})\) by means of individual \((2,2)\) secret-sharing schemes\(^8\) with \( R_{HN,i} \neq R_{HN,j} \) and \( R_{FN,i} \neq R_{FN,j} \) for \( i \neq j \). HN then distributes \( R_{FN,i} \) to \( FN_i \).\(^9\) Unlike with other secret-sharing applications, in our approach, HN keeps copies of \( R_{HN,i} \) as well as the secret roaming key \( R \). This not only allows HN to use the secret roaming key should MD want to access HN directly, but it also enables HN to issue suitable shares to new roaming partners. By construction, \( R \) can be recovered from a collection of shares, if and only if this collection includes a pair \((R_{HN,i}, R_{FN,i})\) for some \( i \in \{1, \ldots, l\} \). In particular, \( R \) cannot be reconstructed from any pair \((R_{HN,i}, R_{FN,j})\) (with \( i \neq j \)) or any collection of shares of foreign networks only. Constructing key pairs with \((R_{HN,i}, R_{FN,i}) \neq (R_{HN,j}, R_{FN,j})\) for \( i \neq j \) is necessary in order to allow for unique identification of \( FN_i \) upon successful authentication.

Authenticating FN by MD using public-key certificates generally involves related operations by both FN and HN, such as decryptions or generating signatures. By introducing the mechanism of key splitting, these operations need to be adapted accordingly. In particular, these operations are now split between HN and FN using distributed decryption or distributed signature generation (see, for example, [40] distributed signatures and [71] additional applications of distributed cryptosystems). Note that the key splitting guarantees

---

\(^8\)See [120] for the definition of a \((2,2)\) secret-sharing scheme.

\(^9\)In standard secret-sharing notation, this corresponds to implementing the access structure \( \Gamma = \{\{HN\}, \{HN_1, FN_1\}, \ldots, \{HN_l, FN_l\}\} \) as an iterative threshold scheme of type \((1,l)(2,2), \ldots, (2,2)\). Unlike in the conventional secret-sharing setting, in our approach the HN, \( \{i = 1, \ldots, l\} \), in fact, do not represent distinct share holders, but the respective shares are all held by HN.
that FN\textsubscript{i} can only be successfully authenticated by MD if HN uses its part R\textsubscript{HN}, of the pair of shares generated for FN\textsubscript{i}. On successful completion of the authentication, HN can thus be sure that the identity FN\textsubscript{i} claimed is correct. This indirectly authenticates FN\textsubscript{i} to HN.

Upon roaming to FN in this new framework, MD now always uses the pre-installed certificate of its HN regardless of FN’s identity. Consequently, MD does not have to validate any certificate. In particular, HN’s participation in a successful authentication automatically confirms the use of a valid roaming key.\textsuperscript{10} Aside from simplifying the handling of certificates considerably, the new key-splitting approach may entail additional advantages over state-of-the-art solutions. These advantages potentially includes a reduction of the number of round-trip message exchanges between MD and HN. Similarly, it may eliminate the need for a secure channel between FN and HN, which, for example, is used to transfer a master key from HN to FN, due to the fact that the key splitting allows FN to use its share to derive the master key from information secured by HN’s share. In Section 11.2 of Chapter 11, we detail the EAP-TLS-KS protocol which implements our new key-splitting approach based on EAP-TLS, and thus enhances WLAN inter-provider roaming. As we will show, EAP-TLS-KS shows all of the above advantages over state-of-the-art roaming solutions.

In summary, the roaming approach sketched here, potentially leads to a roaming authentication and key-agreement protocol of Type 3 that is public-key-based and requires AS\textsubscript{HN}’s online engagement while minimizing the number of round-trips between AS\textsubscript{HN} and AS\textsubscript{FN}. It allows AS\textsubscript{FN} to derive the master key itself and does not require a secure channel. AS\textsubscript{HN} may derive the master key itself, but it is not required to do so. This corresponds to column 11 in Table 2.1. With EAP-TLS-KS, we provided an example roaming authentication and key-agreement protocol that exhibits the claimed potential properties of our approach.

### 2.3 Roaming Across Different Access Technologies

Early generations of wireless devices were equipped with one technology only, limiting inter-operation to providers supporting the same technology. The goal of inter-system roaming support is to enable users to access different networks of different technologies with only one registration process with their home provider and, in the commercial case, only one bill.

As explained before roaming authentication and key-agreement protocols are based on a certain type of credentials. In some wireless technologies, only one particular type of credential is used. For other technologies, roaming authentication and key-agreement protocols based on various credential types are specified. In order to allow MD to roam across different technologies, a home provider has to provide MD with a set of credentials of the right type for at least one roaming authentication and key-agreement protocol for each available technology. A single set of credentials may, however, be sufficient if each candidate technology supports an authentication and key-agreement protocol based on this set of credentials.

\textsuperscript{10}In order to avoid an impersonation attack, it is mandatory for HN to immediately notify all of its MDs of the revocation of the current roaming certificate due to compromise and distribute a new one.
A home provider may want to issue only one set of credentials to its pre-registered users and use this to authenticate the user wherever he may roam. This is the case if a network provider has already invested in an expensive authentication infrastructure and wants to reuse it while entering new roaming agreements with network providers that offer different technologies or while building up a network for another technology itself. Additionally, a network provider may simply avoid troubling his users to obtain an additional set of credentials in order to be able to use a new technology but rather enable him to reuse the one he already has. This may call for new roaming authentication procedures to enable roaming across technologies based on one set of pre-established credentials with the home provider.\textsuperscript{11}

\subsection*{2.3.1 Roaming Agreements, Registering for Roaming and Security Mechanisms}

In order to be able to roam across different technologies, a user registers for a certain set of technologies with his home provider.\textsuperscript{12} The home provider assigns one or more credentials for the user such that the user obtains a suitable set of credentials for at least one (roaming) authentication and key-agreement protocol for each technology for which he registers. The same type of credentials may be usable by different technologies such that the number of technologies may exceed the number of credential types assigned for the user.

During the registration, a user and his home provider furthermore select a set of foreign networks to which MD may roam for each technology. This set may be indirectly defined, for example, by a geographical roaming region. MD and its home provider furthermore determine the set of services MD may use upon roaming for each technology and optionally also exchange information on their roaming policies. In case of commercial providers, MD and the home network additionally agree upon the roaming charges. While set during pre-registration, the roaming region or other parts of MD’s roaming profiles may be changed by the user over time. For example, a user may be able to change his profile making a phone call or making changes to his profile using a web-interface.

The home provider enters into roaming agreements with foreign providers that offer the same and different access technologies, as described in Section 2.1.1.

Roaming authentication and key-agreement protocols for users roaming between different technologies can be defined in exactly the same way as in the inter-provider case previously described (see Definition 2.1.1). FN may support a technology the home provider does not support. $\text{AS}_{\text{HN}}$ here only stands for an authentication server operated by the home provider that does not necessarily act as the authentication server if MD accesses a network operated by the home provider. $\text{AS}_{\text{HN}}$—if engaged in the authentication and key agreement at all—supports the home provider back-end of the roaming protocols for FN’s technology. The same holds for the roaming key-agreement protocol that in the inter-system roaming

\textsuperscript{11}The EAP-Method EAP-SIM [88] for SIM-card-based authentication and key agreement in WLANs is an example for a protocol that was specifically designed to enable roaming across different technologies based on the same set of credentials, namely the secret key shared between MD and its GSM operator.

\textsuperscript{12}Note that the home provider may itself operate networks supporting some of these technologies only.
case can be defined as in Definition 2.1.2. The only difference in the inter-provider case previously described is that MD may roam to a technology its home network may not support.

### 2.3.2 Security Policies

A mobile device roaming across different technologies has to specify a roaming policy for each technology it supports. We denote the technology-specific set of authentication protocols with $RA$, the technology-specific set of key agreement protocols with $RKA$, and the technology-specific sets of key-establishment processes, encryption mechanisms, and integrity-protection mechanisms with $KE$, $EM$, and $IM$ respectively. The set of roaming security suites $RSS$ for this technology is then a subset of the Cartesian product of the sets of technology-specific security mechanisms:

$$RSS \subset RA \times K\Delta \times KE \times EM \times IM$$

This set specifies all technologically possible and standard-wise allowed combinations of security mechanisms for the given technology. We call the elements $rss$ of $RSS$ roaming security suites. MD specifies a subset $RSS_{MD\text{-allow}}$ of $RSS$ that includes all roaming security suites MD allows to be used upon roaming to this technology. Similarly, each FN expresses its policy with respect to the roaming security suites it allows to be used by a subset $RSS_{FN\text{-allow}} \subset RSS$. Finally, the home provider specifies $RSS_{HN\text{-allow}} \subset RSS$ for each technology it has a roaming agreement with a foreign provider for. The home provider may have to set policies for technologies, in particular for cipher suites, he does not support himself. Other than that, the policies described here do not differ from the ones described in the inter-provider case.

Roaming protocols designed for roaming across different technologies can be classified in the same way as in the inter-provider case. Our key-splitting approach can easily be generalized to roaming across different technologies that all support public-key-based roaming authentication without any changes.

### 2.4 Related Issues and Future Directions of Research

#### 2.4.1 Location Update

Upon roaming, MD is often required to be reachable in the same way as if it was connected to its home network (e.g., in the mobile phone case, a roaming mobile device should be reachable for incoming phone calls). This requires MD or FN to acknowledge MD’s current location to HN via a so-called location update procedure after or during authentication. While location update procedures within one technology are relatively easy to achieve, location updates upon roaming across different technologies may be difficult. In particular, a mobile device may use different MAC layer identifiers across different technologies or may change, e.g., its IP address, which makes mappings between different identifiers necessary. A
standard example for location update procedures on the network layer in IP-based networks is Mobile IP [142]. Location update procedures are out of scope of this work.

2.4.2 Intra-Provider Roaming

To avoid frequent authentication and key agreement between users that move within the same network of one provider, some wireless technologies support fast re-authentication mechanisms. If MD was recently authenticated and agreed upon keys with the network and subsequently tries to re-connect to the network, for example, via a new NAP, MD may request fast re-authentication within a certain time period. A mechanism like this is usually based on the master key that was most recently agreed upon between MD and the visited network. If MDs are to be reachable by third parties upon roaming, the fast re-authentication causes the same location update as any full authentication. In case FN operates more than one AS\textsubscript{FN}, the master key may have to be transferred from one AS\textsubscript{FN} to another in order to allow for fast re-authentication. Fast re-authentication mechanisms are, for example, standardized for GSM, UMTS, and WLAN [62, 9, 93]; more recent suggestions in the WLAN context are [129, 137]. In the rest of this work, we concentrate on roaming between different providers and technologies rather then on intra-provider roaming.

2.4.3 Accounting

In the commercial case, additional security issues arise from accounting. As HN reimburses FN for service provisioning, FN has to prove to HN that one of HN’s MDs has indeed used a certain service for a certain amount of time. Accounting can be handled offline or online. Offline accounting is used in older mobile phone technologies. For example, in GSM, so-called Call Detail Records (CDRs) are generated by a visited network\textsuperscript{13} about any chargeable event. These CDRs include information about the user’s identity, the service he used, the duration of the service usage, and MD’s location during service use.\textsuperscript{14} FN collects these CDRs and presents them in a clearing phase to HN. CDRs do not prove MD’s service use to HN, but they provide HN and MD with all available information on the service use, thus allowing MD to repudiate the service usage. Nevertheless, more recently the risk of fraud through malicious service providers has been acknowledged (e.g., [163]) and the integration of non-reputable (online and offline) accounting schemes has been suggested (e.g., [90]) in the mobile phone context.

A detailed discussion of accounting schemes and their usage in wireless access networks in general is out of scope of this work.

2.4.4 Anonymity

State-of-the-art wireless access networks either do not protect MD’s long-term identity (e.g., 802.11) or do protect its confidentiality, but only on the air interface (e.g., UMTS, GSM,\textsuperscript{13} the foreign or the home network\textsuperscript{14} The location is provided in the form of the Cell-ID of the serving BTS.
In particular, the long-term identity of MD is not kept secret from foreign providers to which MD roams. For accounting reasons, the home network needs to get knowledge of a mobile device’s long-term identity, but not the foreign provider. Recently, several solutions to this problem have been suggested (e.g., [21, 188]). However, anonymity issues are out of scope of this work.

2.4.5 Roaming Mediators

As providers aim to offer global coverage for their users, pairwise roaming agreements become infeasible. Instead, so-called roaming mediators (e.g., [15, 49, 166, 29]) come into play. Roaming mediators come in two different functionalities: roaming brokers and delegates. Roaming brokers simply sell packages of roaming partners to a provider. In the commercial case, they additionally take care of clearance and billing between the providers. As opposed to this, roaming delegates handle all security- and mobility-related issues on behalf of a provider, such that the wireless access network provider only provides the physical access network, while everything else is left to the roaming delegate.

2.4.5.1 Roaming Brokers

For providers that aim to offer world-wide coverage to their customers, entering into pairwise roaming agreements with a couple of network providers, e.g., in every country is quite work-intensive. Roaming brokers facilitate the development of these agreements. A network provider does not enter into pairwise (mostly bilateral) roaming agreements, but rather buys a package of roaming partners from a broker or obtains a package of roaming partners. In the case of commercial network providers, the roaming broker is responsible for charging and clearance between the providers.

On roaming, the broker may additionally act as proxy for authentication data requests. Instead of engaging only HN into the authentication on roaming, the broker is involved and forwards authentication traffic between HN and FN. In this case, the broker and HN, as well as the broker and FN, have pre-established credentials they can use to establish a secure channel. If HN and FN have to exchange confidential information during the mutual authentication between MD and FN, these secure channels may be used. It is important to note that in this case, all confidential information is accessible by the broker in the clear.

If the broker does not act as proxy on roaming and the roaming procedure requires a secure channel between HN and FN, then HN and FN have to be enabled by the broker to establish such a secure channel directly. As a future direction of research, the key-splitting approach and other approaches could be extended to support roaming brokers such that no secret information becomes available within the broker and yet no secure channel between FN and HN has to be established. The use of roaming brokers has been described in [15, 49, 166] for mobile phone networks, and in [29] in a more general, public-key-based context.15 However, through the remainder of this work, we consider pairwise roaming agreements only.

15Roaming brokers are referred to as Roaming Service Providers (RSPs) in [29].
2.4.5.2 Roaming Delegates

Offering global or wide-area coverage is particularly difficult for providers who, themselves, only cover a small area. In addition to finding suitable roaming partners and entering into pairwise roaming agreements with them, some small providers will want to delegate all related security issues to a third party. In the commercial case, they will additionally want to delegate the charging and billing for their own as well as foreign mobile devices. This functionality is offered by what we call roaming delegates.

A user does not register with a particular network provider, but rather with the roaming delegate that may itself not operate any access networks at all. The roaming delegate performs all tasks of a home provider, though. All user-related information, including the roaming plan and the exchanged credentials, are stored by the roaming delegate. Upon roaming, the roaming delegate plays the same role as the home network does when pairwise roaming agreements are used. Roaming between different roaming delegates then requires pairwise roaming agreements between the roaming delegates.

Roaming delegates have not been extensively explored in current literature. The authors of [154] take a first step in this direction within the scope of the IST\footnote{Information Society Technologies.} project TORRENT.

2.5 Related Work

**Mobile Telecommunication Standards.** The standardized solutions for inter-provider roaming between different mobile operators supporting the same technology such as two GSM, two UMTS, or two CDMA2000 providers are based on a secret key shared between MD and its home network. The authentication and key agreement requires HNs online engagement on the first of a home provider set number of authentication instances [62, 9, 153]. Subsequent authentications do not then require HN’s engagement any longer. The master key is derived in the home provider and has to be transferred to AS\textsubscript{FN} by means of a secure channel. The home network is not engaged in the cipher-suite negotiation. The roaming authentication and key-agreement protocols used in mobile telecommunication networks are of Type 3 and correspond to the columns six and 12 of Table 2.1 in Section 2.1.5. We will discuss the details of GSM and UMTS inter-provider roaming in Chapter 7.

Patiyoo et al. in [141] suggest reusing the GSM roaming protocols to implement authentication and key agreement upon roaming in wireless ATM\footnote{Asynchronous Transfer Mode.} networks.

**Roaming across WLANs.** In the WLAN context, many different inter-provider roaming security procedures have been proposed. We will discuss these solutions in detail in Chapter 11, where we present our own key-splitting-based roaming authentication and key-agreement protocol for WLAN. At this point, we only mention three WLAN roaming solutions to demonstrate how diverse state-of-the-art solutions are.

Most public WLAN access providers currently use the web-based Universal Access Method (UAM), a method which is also recommended as the best current practice for
inter-provider roaming by the Wi-Fi Alliance [18]. In UAM, a user is authenticated based on a username/password combination in a Type 1 protocol. The properties of UAM correspond to the first column in Table 2.1. Other web-based roaming protocols, like the ones of Bahl et al. or Appenzeller et al. [23, 19], have the same properties.

WLANs that support the new 802.11i security architecture can be set up to support any EAP-Method in combination with a RADIUS server proxy hierarchy [189] on roaming. The roaming authentication and key-agreement protocol based on this setting is then a Type 2 protocol. All EAP-Method-specific authentication traffic is forwarded to the home provider. The home provider’s authentication server authenticates MD in the same way as if MD requested service to it. Depending on the EAP-Method used the authentication can then be public-key-based or be non-public-key-based.

Salagrelli et al. in [156] propose a symmetric-key-based Type 3 roaming authentication protocol that requires $AS_{HN}$ to be online in each authentication and requires a secure channel between $AS_{HN}$ and $AS_{FN}$ to transfer the secret master key.

Roaming across Mobile Telecommunications Technologies.

*GSM-UMTS.* GSM and UMTS are telecommunication standards in which the authentication and key-agreement protocols are based on a shared secret key and home-provider-defined authentication and key-generation algorithms. The secret key and the algorithms are stored on a smart card, called *SIM* in GSM and *USIM* in UMTS. To facilitate the transition from GSM to UMTS mobile devices that support both radio interfaces and are equipped with either type of smart cards, should be able to access both network types. The UMTS standard (see [9]) therefore defines several different authentication and key-agreement protocols for SIM holders roaming to UMTS or USIM holders roaming to GSM. We will discuss all of these procedures in detail in Chapter 8.

*UMTS-CDMA2000.* Kim et al. in [105] explore how roaming of UMTS subscribers to CDMA2000 networks and vice-versa could be implemented. The authors argue that due to major differences in the authentication protocols standardized for these technologies, MDs have to be issued separate credentials for each technology. The home provider of a subscriber is then required to support an authentication server for each technology type.

Roaming across WLAN and Mobile Telecommunication Networks. In order to minimize the cost to offer complementary WLAN access, a recent goal of mobile phone operators is to develop roaming authentication and key-agreement protocols that enable their subscribers to reuse their credentials on accessing WLAN. Examples for authentication protocols like this are the EAP-Methods EAP-SIM and EAP-AKA that are described in [87, 16, 30, 106]. Similar solutions are described in [157] by Salkintzis et al. (for WLAN-GPRS roaming) and by Buddhikot et al. in [43] (for WLAN-CDMA2000 roaming). The basic idea of these protocols is to make the SIM or USIM card used in GSM/GPRS or UMTS reusable in the wireless LAN context. The 3GPP standard [12] specifies the use of EAP-SIM and EAP-AKA.

Chen et al. in [45] suggest a PEAP-based solution in which a UMTS subscriber is authenticated based on its USIM or any other credential type while it authenticates the
WLAN provider based on its public-key certificate. This certificate is provided to MD over the UMTS interface by its currently serving UMTS provider.

While the above solutions require the WLAN provider to support mobile telecommunication network type authentication, the authors of [98] suggest a WLAN-centered approach. Here, the GPRS provider is required to operate a WLAN-type authentication server and subscribers to a GSM/GPRS provider are issued separate WLAN credentials. Upon roaming authentication, all authentication traffic is forwarded to the home provider (Type 2 protocol). Similar suggestions include the use of a regular EAP-TLS implementation with a separate set of credentials for mobile subscribers that roam to WLAN [101]. In this case, the mobile operator’s authentication server has to support EAP-TLS-based authentication and the local authentication server \( AS_{FN} \) forwards all authentication traffic to the home provider.

Tseng et al. [169] propose an authentication protocol for MDs equipped with UMTS credentials. In this protocol, MD communicates with its UMTS home provider over the UMTS air interface and with the WLAN authentication server over the WLAN interface simultaneously. The protocol is based on the UMTS credentials, as well as a hash chain shared between the UMTS and the WLAN provider at some time before the authentication takes place. The UMTS provider generates the master key subsequently used to protect the WLAN air interface. As opposed to the other protocols described so far, MD transfers this master key to the WLAN provider encrypted with the help of a key issued by its home provider for this purpose. Unfortunately, this protocol does not seem to scale, as a UMTS provider has to allocate a different hash chain root for each MD and each WLAN provider. The authors extend this approach in [168] to a public-key-based protocol to support non-repudiatio.

**Technology-Independent Solutions.** In [156], Salgarelli et al. suggest a general roaming authentication framework based on the Needham-Schroeder authentication [132]. The protocol splits the actions taken by one side of the original protocol between the home and the foreign provider in a way that reduces the number of round-trip message exchanges required between \( AS_{HN} \) and \( AS_{FN} \) to one. The suggested roaming protocol is of Type 3, is based on a secret key shared between MD and its home provider, and requires HN’s online engagement in each authentication. The master key is derived in \( AS_{HN} \) and a secure channel is required to transfer this key to \( AS_{FN} \).

In [29], Bayarou et al. suggest a public-key-based approach for roaming users that explicitly addresses the problems regarding the validation of individual certificates on a mobile device. In particular, MDs delegate the validation of a foreign provider’s certificate to a trusted server. However, the authors do not address how changes in roaming profiles or roaming agreements are to be handled. This Type 1 roaming solution supports the first and the third columns of Table 2.1. HN is either not engaged in certificate validations or is engaged and online. Gu et al. [81] also suggest a public-key-based solution, but they do not address the aforementioned inherent problems of such solutions.

In [131], Molva et al. propose a roaming authentication and key-agreement protocol that is based on a secret key shared between MD and HN. It involves HN in each authentication.
The master key is derived in $AS_{FN}$. The protocol nevertheless requires a secure channel between $AS_{HN}$ and $AS_{FN}$. It requires only one round-trip of message exchanges between $AS_{HN}$ and $AS_{FN}$, thus minimizing the overhead on the backbone network.

The IST project SHAMAN specifies a system architecture that allows for roaming across different technologies [72] and specifies design goals for roaming authentication and key-agreement protocols. In particular, the authors argue in favor of using a single set of credentials across different technologies.

In [109], Lin et al. propose an authentication and key-agreement protocol based on a secret key shared between MD and its home provider. Additionally, public-key cryptography is used to protect MD’s permanent identity. In this protocol, $AS_{HN}$ is required to be online and derives and transfers the secret master key to $AS_{FN}$.

**Related Secret-Sharing Applications.** The use of key-splitting to facilitate revocation of certificates used by users for signing or encryption purposes has been described in [38]. We make use of this idea in the sense that key-splitting of a roaming key makes roaming agreements easy to revoke.

Geer and Yung suggest alternative applications of threshold cryptography in [71]. Although this work does not address inter-provider roaming, this paper inspired our work.

### 2.6 Conclusion

Public-key-based methods, in principle, have the advantage that a foreign network and MD can authenticate each other without the involvement of MD’s HN. However, most roaming scenarios, particularly commercial ones, require that every instance of roaming be controlled by HN. Therefore, most public-key-based methods and all non-public-key-based methods suggested for roaming to date require that the mutual authentication involves HN. HN authenticates MD and assures FN of MD’s authorization to roam to FN. Similarly, HN authenticates FN and assures MD of FN’s authorization to offer service to MD.

Furthermore, in authentication methods that require HN’s interaction, the cryptographic keys used for MAC layer protection (between MD and FN) following a successful authentication are typically generated by HN. This requires HN to establish a secure channel to FN in order to allow for a secure transfer of these cryptographic keys.

Another common shortcoming of public-key-based authentication methods for roaming users is that MD must check the validity and revocation status of certificates during network authentication before actually having network access.

In this chapter, we have modeled and classified roaming authentication and key-agreement protocols and introduced a new secret-sharing-based approach to inter-provider and inter-system roaming. Our public-key-based approach requires HN interaction on each roaming instance, allows FN to derive the cryptographic key material itself, and allows a timely accommodation of changes in roaming agreements and roaming profiles. Our new approach supports pairwise roaming agreements between wireless access network providers. This is the type of agreements used to date by mobile phone operators and wireless internet service providers. An interesting future direction of research might be to extend our
approach to newly evolving types of handover agreements arising from the introduction of roaming brokers and delegates to scene.
Part II

Handover
Part II in the General Context

Part II is the main part of the thesis. In Chapter 3, we model inter-provider handover procedures based on Part I. Handover of mobile devices from their home network to other networks are modeled based on the security model presented in Chapter 1 and handover of mobile devices from a foreign network—to which they initially roamed—to other networks are modeled based on the roaming model presented in Chapter 2. In Chapter 4, we analyze the threats imposed on mobile devices and wireless access networks by inter-provider handover procedures that use security-context transfer and derive new security requirements. In Chapter 9 of Part III, we analyze the intra-system and inter-system handover procedures between GSM and UMTS networks based on the newly derived security requirements. In Chapter 5, we present the new history-enriched, policy-based approach to inter-provider handover. This approach is further detailed in the WLAN context in Chapter 12 of Part IV. In Chapter 6 we extend the new handover approach to inter-system handover.
Chapter 3

Inter-Provider Handover—Model Procedures and Security Solutions

In a wireless access network, a mobile device is connected to a NAP by means of a radio link. This network access point has a transmission range that is usually referred to as a cell. A handover procedure generally allows a user to move from one cell to another without loss or even disruption of the services he currently uses. During a handover procedure the mobile device of the user switches from a (MAC layer) connection with one NAP (the source NAP) to another one (the destination NAP). On the network side, a handover procedure makes the re-routing of incoming and outgoing data traffic over the destination NAP necessary.

In the simplest case, a handover takes place within the network of one network provider. The source and the destination NAP belong to the same wireless access network and are connected to the same backbone. This type of handover is referred to as intra-provider handover. Intra-provider handover are implemented in all current mobile communications networks and have lately also been specified for IEEE 802.11 WLANs [60, 11, 91, 93].

The coverage of a single wireless access network provider is usually restricted to a certain geographical area. Handover procedures between different providers (inter-provider handover) aim to offer seamless services to users in a wider coverage area than the one offered by the home provider. On an inter-provider handover, the source and the destination NAP belong to different wireless access networks, the source network (SRC) and the destination network (DEST) of the handover procedure. Consequently, they are connected to different backbone networks. However, these two backbone networks are required to be interconnected in order to allow for the re-routing of data traffic. Handover procedures between different providers, though widely discussed, as for example in [162, 177, 27, 184, 152], are currently very rare. One of the few examples for already standardized inter-provider handover is the handover procedures between a GSM provider and a UMTS provider [11]. Inter-provider handover are expected to be regulated via handover agreements between network providers that regulate the mobility management, security-related issues, and in the

\[1\] Some authors (e.g., [29, 177]) prefer the terms inter-domain and intra-domain handover and use them in the same way as we use the terms inter-provider and intra-provider handover.
case of commercial providers, the terms for accounting and billing. Handover agreements are expected to play a similar role in the wireless network access provider world as roaming agreements do nowadays in mobile communications networks [1]. Users then register for handover to a certain set of foreign providers (e.g., providers in a certain geographical region) as part of the registration process with their home providers.

Upon inter-provider handover, a mobile device starts to use a service while it is connected to a certain network access point. This network access point is called the anchor NAP of the handover procedure. We call the network the anchor NAP belongs to, the Anchor Network (AN) and the provider that operates the anchor network, the anchor provider. A handover from AN as SRC network to another network DEST is referred to as first-order handover here. A first-order handover can take place within the network of one provider or between two different providers. Moreover, the two networks AN=SRC and DEST can support the same or different technologies (inter-system handover). A handover from the destination network DEST_1 of a first-order handover to yet another network DEST_2 is referred to as second-order handover and so on. Handover of an order greater than one are also-called subsequent handover. Subsequent handover can in particular differ from first-order handover in the mobility management on the network side. The re-routing of traffic to the destination NAP can be handled by the source network of the handover, by the home network of the mobile device, or by the anchor network.

The anchor network on an inter-provider handover may be the home network of the mobile device. However, as more and more roaming procedures between different providers and wireless network technologies evolve, handover procedures of a mobile device from a foreign network to another network become of greater interest. Consequently, the anchor network may either be HN or any FN that has a roaming agreement with HN.

Figure 3.1 illustrates the order and anchor type of a handover and clarifies the numbering we use throughout the remainder of this work.

The security challenge on inter-provider handover is that MD and the destination network have to mutually authenticate each other, negotiate a cipher suite to use after handover, and establish a master session key without causing a disruption of any ongoing connections.

In this chapter, we model various types of first-order and subsequent inter-provider

---

**Figure 3.1: Order and Anchor Type of a Handover**
handover procedures within the same technology. Inter-system handover are subsequently studied in Chapter 6. Our model is based on handover procedures used in state-of-the-art wireless access networks, as well as other recently published work. In particular, the modeling is based on the procedures used in UMTS [11], GSM [60], CDMA2000 [13], and IEEE WLAN [91, 92], as well as the procedures modeled and described in [182, 127, 135, 17, 176]. The derived model procedures are not only an aggregated summary of current work in the field. For once, the model procedures are described in a technology-independent way. As a consequence, they are more general than the aforementioned work. More importantly, previous work on inter-provider handover does not differentiate between first-order and higher-order handover. Consequently, subsequent inter-provider handover procedures are not explicitly addressed by previous work. By introducing the anchor network and the subsequently serving source and destination networks, our model explicitly addresses subsequent handover and handover after roaming.

Furthermore, we describe the security challenge imposed by first-order and subsequent inter-provider handover and briefly summarize and discuss state-of-the-art approaches to address this challenge. The first approach is based on an authentication and key agreement between MD and DEST\textsubscript{k} during handover, e.g., the same protocols as used upon roaming of MD to DEST\textsubscript{k}. We discuss this approach mainly to motivate the need for other solutions. The second approach generalizes the so-called pre-authentication method suggested in [137] for intra-provider handover in WLAN and adopted for the new security architecture 802.11i [93] to the inter-provider case. The third approach, Security-Context Transfer (SCT) with key derivation, generalizes the solutions currently used to support intra-provider handover, e.g., in GSM [60], UMTS [11], CDMA2000 [13], and adapted to WLAN ([129, 185]) to the inter-provider case. In this approach, the master session key used after handover is derived from previously used master session keys. Although the use of SCT with key derivation has previously been suggested in the inter-provider context by [162, 75, 186, 27, 177], none of this previous work explicitly addresses subsequent handover or distinguishes between handover with HN and FN as anchor. The fourth security solution finally combines the second solution with the efficiency advantages of the third and uses SCT with key agreement. As opposed to SCT with key derivation, the master key used after handover is agreed upon based on the credentials exchanged between MD and HN as part of the pre-registration process and is not derived from any previously used master session keys.

In a brief discussion of the advantages and disadvantages of the above four security solutions, we identify methods using security-context transfers during handover as the most promising approach to meet the efficiency requirements that seamless handover impose.

**Outline.** In Section 3.1, different types of handover procedures are modeled. This is followed by a description of the security challenge on inter-provider handover in Section 3.2 and a brief discussion on different solutions in Section 3.2.1 to Section 3.2.4. Section 3.3.1 details the relation between our model procedures and the aforementioned work in this field. In Section 3.3.2, we provide a more detailed treatment of the related work on security solutions for inter-provider handover.
3.1 Model Procedures

A handover procedure consists of several phases. In the first phase, a handover reason is detected. The obvious reason to initiate a handover procedure for a mobile device is that MD moves out of the transmission range of the currently serving NAP. This handover reason is a so-called imperative reason (see [187]), as the handover must take place in order not to lose connectivity. Other handover reasons are so-called alternative handover reasons. An alternative handover reason indicates that a handover is desirable but not required for a seamless use of service. An alternative handover reason, for example, occurs if a mobile device moves into the range of a network access point that is preferable to the current one because of a stronger signal while the currently serving NAP is still available. Another alternative reason to initiate handover is load balancing: MD is in the transmission range of more than one NAP (cell intersection or overlay) and the currently serving NAP is overloaded. A third example for an alternative handover reason is somewhat opposite to load balancing: a mobile device aims to be connected to the closest network access point to save battery power by reducing the necessary control power (see [184]).

The detection of a handover reason is based on a so-called handover algorithm. An overview on basic handover algorithms can be found in [83]. A handover algorithm takes collected measurement data as input and outputs whether or not a handover should take place. This measurement data typically includes the currently received signal strength, the current load on the serving NAP, the signal to interference ratio, the bit-error rate, the carrier interference ratio, etc. [187]. Throughout the remainder of this work, the actual handover algorithm is of no interest. We just assume that at some point a handover reason is detected by some specific handover algorithm.

Once a handover reason is detected, a new NAP, the destination NAP is selected in the second phase of a handover procedure. The choice of the destination NAP typically depends on the signal strength of the NAPs and whether these NAPs have the free capacity to serve MD after handover. We do not further specify the non-security-related part of this decision process in this work.

In the third and last phase, the execution phase, MD disassociates from its currently serving NAP, the source NAP of the handover, and connects to the destination NAP. The execution phase also includes the mobility management on the network side that guarantees the re-routing of incoming and outgoing data traffic over the new network access point.

Handover procedures can be classified into mobile-initiated, network-initiated and mobile-assisted handover procedures. In a mobile-initiated handover procedure, MD detects handover reasons, while in the network-initiated case, the currently serving network detects handover reasons. In a mobile-assisted handover procedure, MD provides the network with measurement data on the reception level of surrounding NAPs. The network processes this measurement data in order to determine handover reasons. Depending on who selects the destination network and initiates the execution of the handover, handover procedures are further classified into mobile-controlled and network-controlled handover (see [116]). For a detailed discussion of network-controlled versus mobile-controlled handover procedures, we
3.1 Model Procedures

refer to [184]. It is, however, important to note that security issues are not addressed in [184].

Independent of the type of initiation and the control type, handover procedures are further classified into hard and soft procedures.\(^2\) In hard handover procedures, MD can only be connected to one NAP at a time. In the execution phase, MD consequently first disassociates from the source NAP before it associates with the destination NAP, thus causing a disruption of the incoming and outgoing data traffic. In order to provide seamless use of service, hard handover procedures have to be fast, such that the disruption does not result in a disruption of the services used. In contrast, in a soft handover procedure MD, associates with several NAPs at a time. In the execution phase, MD is first connected to the destination NAP only, then connected to both the source and the destination NAP for some time before it disconnects from the source NAP. Soft handover procedures have the advantage that data traffic to and from the mobile device can be sent to and received from both the source and the destination NAP, as long as the mobile device is connected to both of them. Consequently, soft handover procedures can easily support uninterrupted service use. Hard handover procedures are, for example, standardized for GSM and IEEE 802.11 [60, 91, 93], while soft handover procedures are standardized for UMTS and CDMA2000 [11, 13].

In this section, we present a new model for the various types of inter-provider handover procedures that explicitly models subsequent handover. In particular, we model hard and soft mobile-assisted, network-initiated handover procedures based on [60, 11, 17, 13], as well as hard and soft mobile-initiated, mobile-controlled handover procedures based on [176, 182, 127]. To facilitate the reference to the modeled handover procedure types we, throughout the remainder of this work refer to mobile-assisted, network-controlled handover procedures simply as network-initiated handover procedures and to mobile-initiated, mobile-controlled procedures simply as mobile-initiated handover procedures. In both the network-initiated and the mobile-initiated cases, we start by modeling first-order handover from HN to a foreign network (FN). Then, we model subsequent handover with HN as the anchor network. Each case ends with the modeling of first-order and subsequent handover with FN as the anchor network.

Figure 3.2 illustrates an inter-provider handover in general. A mobile device is currently associated with one of the network access points of some source network SRC and uses a service. When a handover reason is detected, the mobile device moves out of the coverage range of SRC and the mobile device is handed over to some DEST. SRC and DEST support the same technology but are operated by different network providers.

\(^2\)Some authors (e.g., [184, 116]) use the terms make-before-break to denote a soft handover and break-before-make to denote hard handover.
3.1.1 Hard and Soft Network-Initiated Handover

3.1.1.1 First-Order Handover with HN as Anchor

Figure 3.3 models a network-initiated first-order handover procedure of MD from its HN to some DEST. In a network-initiated handover, the network collects measurement data related to the quality of the link layer connection between MD and the currently serving NAP. This measurement data typically includes the currently received signal strength, the current load on the serving NAP, the signal to interference ratio, the bit-error rate, the carrier interference ratio, etc. [187].

While the network may measure parts of the data itself, MD can also assist in this task (mobile-assisted handover). In this case, the mobile device measures its reception level of the surrounding network access points, including the currently serving one. It sends measurement reports to HN and the HN processes these reports.

Based on the collected and processed measurement data, HN detects a handover reason. The detection of a handover reason is based on a so-called handover algorithm. An overview on basic handover algorithms can be found in [83]. A handover algorithm takes the collected measurement data as input and outputs whether or not a handover should take place. Throughout the remainder of this work, the actual handover algorithm is of no interest. We

---

The measurement data may also include some estimate on the current location of the mobile device. If the network component collecting the measurement data has knowledge of the network topologies of its own and the surrounding networks, then knowing the location of the mobile device means knowing whether or not the mobile device is in the transmission range of other network access points of the home network or foreign networks.
just assume that HN at some point detects a handover reason using some provider-specific handover algorithm.

HN subsequently uses the collected measurement data to generate an ordered list $L = \{\text{DEST}^1, \ldots, \text{DEST}^n\}$ of candidate destination networks for handover.\footnote{This list also specifies the network access points in these candidate destination networks.} We do not make any assumptions about the algorithm HN uses to determine the list of candidate networks here. We only assume the existence of such an algorithm. As long as MD receives a NAP belonging to HN with sufficient quality and with free capacity, HN itself will be DEST$^1$ in $L$ and HN initiates an intra-provider handover. However, in this chapter we concentrate on inter-provider handover. Intra-provider handover will briefly be discussed in Section 6.3. In this section, we assume that all destination networks in $L$ are foreign networks. HN chooses the candidate destination network DEST$^1$ in $L$ and sends a handover request to it. This handover request includes the identities of MD, its HN and DEST$^3$. It may also include a list of allowed subsequent handover destinations and other security-related information.

Upon receiving a handover request, DEST$^1$ decides whether or not to accept the handover request and answers accordingly with a positive or negative handover response. If DEST$^1$’s handover response is positive, HN sends a handover command to the mobile device commanding handover to DEST$^1$. If the handover response of DEST$^1$ is negative, HN tries the candidate destination network DEST$^2$ from its list until it receives a positive answer from the $i$-th candidate network DEST$^i$ in $L$. HN then selects DEST$^i$ as the next destination network DEST and sends a handover command to MD including DEST = DEST$^i$’s identity. If none of the destination networks in $L$ sends a positive handover response, a handover is
not possible and HN has to drop the connection.

In the case of a hard handover procedure, MD disassociates from HN as soon as it receives a handover command to a destination network DEST. In a hard handover procedure, HN typically keeps resources (for example, a channel) allocated for MD until it receives a handover complete message from the destination network, indicating successful handover. MD can fall back to its old NAP in HN if it is still in its range. As soon as MD has successfully associated with DEST, DEST sends a handover complete message to HN. If the association fails, MD tries to re-associate with HN.

In the case of a soft handover procedure, MD upon reception of a handover command first associates with DEST. MD disassociates from HN if and only if it has successfully associated with DEST.

In order to model as technology-independent as possible, we do not further specify which components within HN or DEST control the handover procedure.

In the following section, we generalize the above handover procedure to subsequent handover and handover after roaming.

### 3.1.1.2 k-th-order Handover with HN as Anchor

A mobile device has established a connection with its HN and started to use a service of some service provider. MD has subsequently been handed over from HN = SRC\textsubscript{1} to a destination network DEST\textsubscript{1} by means of the first-order handover procedure described in the last section. After \( k - 1 \) subsequent handover from SRC\textsubscript{i} (2 ≤ i ≤ k − 1) to DEST\textsubscript{i} (2 ≤ i ≤ k − 1), MD is connected to the destination network DEST\textsubscript{k−1} of the (k − 1)-st-order handover. DEST\textsubscript{k−1} is the source network SRC\textsubscript{k} of the k-th-order handover. A new authentication between HN and MD resets the handover chain to HN = SRC\textsubscript{1}.

We distinguish between two control types for subsequent handover procedures with HN as anchor: HN-controlled handover and SRC-controlled handover. These control types reflect different types of handover agreements between the wireless access networks.

**HN-Controlled Subsequent Handover.** In an HN-controlled k-th-order handover, the source network SRC\textsubscript{k} determines that a handover reason occurred and informs HN. HN selects the candidate destination network and initiates and authorizes the actual handover. MD accepts handover commands that originate from its HN only. A handover to DEST\textsubscript{k} can take place if HN and DEST\textsubscript{k} have a handover agreement. It is HN that assures DEST\textsubscript{k} that MD is authorized to be handed over to DEST\textsubscript{k}. In the case of commercial providers, HN’s authorization provides DEST\textsubscript{k} with the guarantee that HN will reimburse DEST\textsubscript{k} for its service provisioning.

Figure 3.4 details an HN-controlled k-th-order handover procedure. The source network SRC\textsubscript{k} of a k-th-order handover sends a handover indication to HN as soon as it detects a handover reason. SRC\textsubscript{k} includes the necessary measurement data in the indication to give HN the power to process it. HN proceeds as in the first-order handover case with the generation of a list of candidate destination networks and sends handover requests to them. Upon receiving a positive handover response from one of them, HN sends a
3.1 Model Procedures

handover command to SRC\textsubscript{k} including the identity of the selected DEST\textsubscript{k}. SRC\textsubscript{k} forwards the handover command to MD.

![Figure 3.4: HN-Controlled Subsequent Handover](image)

In the case of HN-controlled handover, the handover agreement between two providers \( P_1 \) and \( P_2 \) consists of two parts, one that regulates handover of \( P_1 \) pre-registered MDs to \( P_2 \) (HN = \( P_1 \), DEST = \( P_2 \)) and one that regulates handover of \( P_2 \) pre-registered MDs to \( P_1 \) (HN = \( P_2 \), DEST = \( P_1 \)).

**SRC-Controlled Subsequent Handover.** In SRC-controlled \( k \)-th-order handover, the source network SRC\textsubscript{k} determines the handover reason, selects the candidate destination network, and initiates and authorizes the actual handover. A handover of MD from SRC\textsubscript{k} to DEST\textsubscript{k} can take place if SRC\textsubscript{k} and DEST\textsubscript{k} have a handover agreement. Note that in the case of an SRC-controlled handover, the handover agreement between two providers regulates handover of any MD from SRC\textsubscript{k} to DEST\textsubscript{k} such that the agreement is not restricted to their respective pre-registered MDs. The HN of MD is only involved in the \( k \)-th-order handover procedure if it is the source or destination network. Consequently, HN delegates the control of a second-order handover to DEST\textsubscript{1} = SRC\textsubscript{2}; SRC\textsubscript{2} delegates control of a third-order handover to DEST\textsubscript{2} = SRC\textsubscript{3}, and so on. MD is assured of SRC\textsubscript{k}'s authorization of the handover as soon as it receives a handover command message from SRC\textsubscript{k}. Similarly, SRC\textsubscript{k}'s authorization of the handover is assured to DEST\textsubscript{k} in the form of a handover request. In the case of commercial network providers, SRC\textsubscript{k}'s authorization includes its guarantee to reimburse DEST\textsubscript{k} for service provisioning. For a mobile device, a SRC-controlled handover implies transitive trust in the network providers: MD trusts HN’s authorization by means of the initial authentication between MD and HN. Subsequently,
MD trusts handover commands received from HN = SRC₁, DEST₁ = SRC₂ and so on.

An SRC-controlled subsequent handover procedure can be described by replacing HN with SRCₖ in Figure 3.3. In the SRC-controlled case, SRCₖ determines the list of candidate destination networks itself. SRCₖ sends a handover request for MD to the candidate destination networks and commands MD to associate with DESTₖ upon receiving a positive handover response from a destination network.

3.1.1.3 FN as Anchor

On inter-provider handover after roaming, MD first roams to a foreign network FN and starts to use a service via FN. The last authentication and key agreement has taken place while MD was connected to FN. If a handover reason is detected by FN, a first-order inter-provider handover with FN as anchor is initiated. The selection of the destination network and the initiation of the first-order handover itself can be controlled either by FN or by HN.

HN-Controlled Handover. In the HN-controlled case, HN selects the destination network, and initiates and authorizes the actual handover. An HN-controlled first-order handover with FN as anchor can be described by replacing SRCₖ with FN in Figure 3.4.

FN-Controlled Handover. In the FN-controlled case, FN selects the destination network, and initiates and authorizes the actual first-order handover. An FN-controlled first-order handover with FN as anchor can be illustrated by replacing HN with FN in Figure 3.3.

FN-Controlled Subsequent Handover. Subsequent handover with FN as anchor result in a chain of subsequently serving networks: FN = AN = SRC₁, DEST₁ = SRC₂, . . . . A full (roaming) authentication between MD and FN or between MD and any other FN or HN resets the handover chain.

Subsequent handover with FN as anchor can be SRC-controlled, HN-controlled, or FN-controlled. Subsequent FN-controlled procedures with FN as anchor can be described by replacing HN with FN in Figure 3.4. In this control type, FN selects each subsequent destination network, and initiates and authorizes the actual subsequent handover.

HN-Controlled Subsequent Handover. Subsequent HN-controlled handover with FN as anchor are the same as subsequent handover with FN as anchor and can be described by the same procedure (see Figure 3.4). In this case, the source network SRC₁ of a first-order handover is FN. HN selects the destination network DESTₖ of a subsequent handover, and initiates and authorizes the actual handover.

SRC-Controlled Subsequent Handover. Subsequent SRC-controlled handover with FN as anchor are executed in exactly the same way as subsequent SRC-controlled handover
with HN as anchor and can be modeled by replacing HN with SRC\textsubscript{k} in Figure 3.4. SRC\textsubscript{k} selects the destination network DEST\textsubscript{k} of a subsequent handover, and initiates and authorizes the actual handover.

### 3.1.1.4 Summary and Generalized Procedure:

In summary, we distinguish three different types of handover control: HN-controlled handover procedures, SRC-controlled handover procedures, and AN-controlled handover procedures.

**HN-Controlled Handover.** In an HN-controlled handover, HN selects the destination network and initiates and authorizes each \( k \)-th-order handover \((k \geq 1)\).

**SRC-Controlled Handover.** In an SRC-controlled handover, the source network SRC\textsubscript{k} of a \( k \)-th-order \((k \geq 1)\) handover selects the destination network DEST\textsubscript{k} and initiates and authorizes the actual handover. Consequently, HN delegates the control of a first-order handover to AN (where AN is HN or any FN that has a roaming agreement with HN). AN = SRC\textsubscript{1} delegates the control of a second-order handover to DEST\textsubscript{1} = SRC\textsubscript{2} and so on.

**AN-Controlled Handover.** In an AN-controlled handover, all \( k \)-th-order handover \((k \geq 1)\) are controlled by AN. AN selects the destination network DEST\textsubscript{k} and initiates and authorizes the actual handover. Consequently, in an AN-controlled handover, HN controls subsequent handover with HN as anchor and FN controls subsequent handover with FN as anchor.

Throughout the rest of this work, we assume that all handover procedures specified for a wireless technology are of exactly one control type. That is, we do not study the effect of the subsequent use of handover procedures of different control types.

The different control and anchor types can be summarized by the general network-initiated handover procedure illustrated in Figure 3.5. In this figure, HCN stands for Handover Controlling Network and is to be replaced by SRC, HN, or AN, depending on who controls the handover.

### 3.1.2 Hard and Soft Mobile-Initiated Handover

#### 3.1.2.1 HN as Anchor—First-Order Handover

In the mobile-initiated case, MD detects a handover reason and selects the destination network. There are two different approaches to notify HN of the upcoming handover. Either MD sends a notification message to HN before it associates with the destination network or DEST sends a notification message to HN after MD associates with it. Figures 3.6 and 3.7 illustrate these two cases.
In both cases, the handover procedure is initiated when MD detects a handover reason. MD measures the signal strength and other quality-of-service indicating parameters of its currently serving NAP, as well as other available NAPs. As a result, MD generates an ordered list of candidate destination networks by an algorithm we do not further specify here. HN can assist MD in generating the list of candidate networks. For example, HN may send a list of candidate destination networks to MD at any time before a handover reason is detected.

In case MD notifies HN of the upcoming handover (Figure 3.6), MD sends a handover-indication message to HN including the selected DEST’s identity right after selecting DEST from the list of candidate destination networks. HN decides whether or not to allow the handover and indicates its authorization to DEST in the form of a handover-indication message that includes the identity of MD.

In the case of a hard handover procedure, MD then disassociates from HN and tries to associate with DEST. After successful association, MD sends a handover request to DEST and DEST answers with a handover-response message indicating its decision to MD.

In the case of a soft handover procedure, MD disassociates from HN only after receiving a positive handover response from DEST.

In the soft and hard handover cases, MD sends a handover request to the next destination network in its list if the handover response of DEST is negative. If none of the candidate networks in $L$ sends a positive handover response, no handover is possible and MD drops the connection with HN.

In case HN is notified by DEST (Figure 3.7), MD tries to associate with the first destination network of its list of candidates. MD then sends a handover-indication message to DEST. If DEST is not willing to accept the handover, it immediately sends back a negative handover response to MD. Otherwise, DEST forwards MD’s handover indication to HN.
HN replies positively or negatively in its handover-request message to DEST and DEST in turn sends its handover response to MD.

In a hard handover procedure, MD disassociates from HN before associating with DEST. In a soft handover procedure, MD disassociates from HN only after receiving a positive handover response from DEST.

3.1.2.2 HN as Anchor—Subsequent Handover

As in the network-initiated case, subsequent handover with HN as anchor network can either be controlled by HN, in which case HN is engaged in every handover procedure, or be SRC-controlled. In the latter case, a subsequent handover procedure can be illustrated by replacing HN with SRC$_k$ in Figure 3.7 and Figure 3.6.

In an HN-controlled handover, DEST$_k$ sends an additional handover indication to HN, to which HN replies with a positive or negative handover request.

3.1.2.3 FN as Anchor—Subsequent Handover

As in the network-initiated case, a subsequent handover with FN as anchor can either be HN-controlled, FN-controlled, or SRC-controlled. The source network SRC$_k$ forwards MD’s handover notification to HCN or DEST$_k$ notifies HCN after MD and DEST$_k$ have associated.
As in the network-initiated case, all of the aforementioned procedure types can be described in a general way as illustrated in Figures 3.8 and 3.9. HCN is to be replaced by HN, SRC, or AN, depending on who controls the handover.

Both types of mobile-initiated inter-provider handover procedures have previously been suggested for first-order inter-provider handover. Oyoqui et al. in [135] present a handover procedure in which HN is notified by MD, while Mishra et al. in [127] and Xhafa et al. in [182] use handover procedures in which HN is notified by DEST. However, none of the previous work generalizes the procedures to the subsequent handover case and identifies the possible handover control types.

3.2 The Security Challenge and Solutions

The security challenge on a \( k \)-th-order handover procedure is that MD and DEST\(_k\) have to be mutually assured of HCN’s timely authorization of the handover instance, negotiate
3.2 The Security Challenge and Solutions

a cipher suite $cs_k = (ke_k, em_k, im_k)$ to be used after handover, and establish a master session key $K_k$ from which they can derive data-protection keys $IK_k$ and $EK_k$ to protect the connection between MD and EIPE$_{DEST_k}$.

As such, the security challenges upon handover and roaming are very similar, with the exception that HCN authorizes handover instances while roaming instances are authorized by HN. However, handover procedures have to additionally fulfill very tight efficiency requirements, arising from the following facts:

1. Services, in particular real-time services like voice connections or video streaming, are sensitive to disruptions such that services have to be reestablished after longer disruptions. Providing seamless real-time services is the most promising application of handover procedures.

2. Users are sensitive to short disruptions of the services they use. While disruptions of services like an HTTP connection can be acceptable for a user, disruptions of voice connections and music or video streaming are not acceptable and are critical for user acceptance of these services. For handover procedures of, for example, voice connections, the ITU advises a disruption time\(^5\) of less than 50 ms [69].

3. In order to provide seamless service use, a handover procedure has to be completed before MD leaves the cell of NAP$_{SRC_k}$. How fast a handover procedure for a given

\(^5\)Here, disruption time refers to the time between the last frame received over the source NAP and the first frame received over the destination NAP. A disruption like this only occurs on hard handover procedures.
wireless technology has to be thus also depends on the size of the cells and their intersections, as well as the speed with which MD moves through the cells. These efficiency requirements call for solutions of the security challenge on inter-provider handover that minimize the introduced service disruption (hard handover) as well as the overall handover latency\(^6\) (hard and soft handover procedures).

In the following sections, we present and discuss four different security solutions. The first one is based on a new run of an authentication and key agreement between MD and DEST\(k\) during handover and is discussed mainly to motivate the need for new solutions. The second one generalizes the so-called pre-authentication method suggested in [137] for intra-provider handover in WLAN to the inter-provider case.\(^7\) The third solution, Security-Context Transfer (SCT) with key derivation, generalizes the solutions currently used to support intra-provider handover in GSM [60], UMTS [11], CDMA2000 [13], and WLAN [129, 185] to the inter-provider case. Although the use of SCT with key derivation has previously been suggested in the inter-provider context by [162, 75, 186, 27, 177], none of this previous work explicitly addresses subsequent handover or distinguishes between handover with HN and FN as anchor. Consequently, none of them identifies and discusses the differences between HN-controlled, AN-controlled, and SRC-controlled subsequent handover that we present here. Finally, the fourth security solution combines the second solution with

---

\(^6\)The overall handover latency is the time between the detection of a handover reason and the completion of the handover.

\(^7\)This solution was included in the new security architecture IEEE 802.11i [93] for WLAN.
the efficiency advantages of the third and uses SCT with key agreement, and generalizing previous intra-provider solutions ([93, 136]) in a way that, to the best of our knowledge, has not been suggested so far.

### 3.2 The Security Challenge and Solutions

3.2.1 Full Authentication Between MD and DEST$_k$ via NAP$_{DEST_k}$

MD and DEST$_k$ negotiate a security suite, authenticate each other, and agree upon a master session key in the same way as upon roaming before the mobility management redirects data traffic to MD over NAP$_{DEST_k}$.

Figure 3.10 illustrates how this solution can be integrated into a network-initiated handover procedure. On association during handover, MD and DEST$_k$ negotiate the security suite to use and then mutually authenticate each other by means of the negotiated authentication protocol $(r)a_k$, agree upon a master session key $K_k$ by means of the key agreement $(r)ka_k$, establish fresh data-protection keys by means of $ke_k$, and subsequently use the agreed-upon encryption and integrity-protection mechanisms $em_k$ and $im_k$ to protect data and control traffic exchanged between them. In the remainder of this section, we refer to this sequence of protocols and processes as MD and DEST$_k$ establish a new secure connection. Note that in this security solution, it is HN that authorizes the handover as part of

![Figure 3.10: Full Authentication via DEST$_k$ in the Network-Initiated Case](image)

the initial authentication. However, the selection of the next destination network DEST$_k$ of a $k$-th-order handover and the initiation of the actual handover can either be controlled by HN, AN, or SRC$_k$.

Upon a mobile-initiated handover, the establishment of a new secure connection can be
integrated into the association between MD and DEST_k the same way as detailed for the
network-initiated case. Therefore, we do not explicitly detail the mobile-initiated case here.

In case a technology supports hard handover procedures only, the establishment of the
new secure connection via NAP_{DEST_k} has to take place after MD has disassociated from
SRC_k. Consequently, the time required to establish a new secure connection causes a
service disruption. As mentioned above, disruptions of real-time traffic are recommended
to stay below 50 ms. However, authentication protocols of state-of-the-art wireless access
technologies require several seconds. For example, in GSM, MD is required to be able to
send an authentication response no more than one second after receiving an authentication
request [61]. A GSM authentication and key agreement thus takes more than one second to
complete. As a consequence, the above method is commonly assumed to be too inefficient
to support seamless use of real-time services in combination with hard handover procedures
(see, e.g., [27, 177, 162]).

In case a technology supports soft handover procedures, MD can be associated with
NAP_{SRC_k} and NAP_{DEST_k} at the same time. Consequently, on soft handover, the establish-
ment of a new secure connection can take place via NAP_{DEST_k} before MD disassociates from
NAP_{SRC_k} such that it does not add to the disruption time. Even in the soft handover case,
a secure connection has to be established as long as MD is in the intersection of the cells
of NAP_{DEST_k} and NAP_{SRC_k}. Consequently, the intersection between the cells of NAP_{DEST_k}
and NAP_{SRC_k} has to be sufficiently large and MD has to move sufficiently slow through the
intersection.

As cell sizes and intersections differ greatly between wireless technologies, providers, and
even environmental circumstances, a general statement on the relation between the size of
the cell intersection, the velocity of MD, and the overall handover delay, is not possible.\(^8\)

Whenever the establishment of a new secure connection in the above described way is
possible, the connection between MD and DEST_k after handover is protected in exactly the
same way as on roaming to DEST_k. The security of the new connection does not depend
at all on the security of the connection before handover. The HN has as much control over
each handover instance as the initial authentication and key agreement allow for and as
much influence on the security-suite negotiation as upon roaming.

The only new security threats imposed by handover procedures that use a full authen-
tication over NAP_{DEST_k} compared to roaming and accessing HN are new Denial of Service
(DoS) attacks that exploit the handover messages and handover-specific behavior of MD,
DEST, HCN, and SRC.\(^9\)

In summary, establishing a new security context via NAP_{DEST_k} is a good choice to
secure soft inter-provider handover in case MDs can be expected to move with low velocity.
However, in some use cases, like, mobile and video telephony, or video streaming in public
transportation or cars, MDs move at relatively high speed. In order to guarantee seamless

\(^8\)For example, assume MD takes a path of length 100 m through the intersection of two UMTS cells
operated by different providers and MD moves at 100 km/h, then MD stays in the cell intersection for only
3.6 s.

\(^9\)The attacks are described in more detail in Section 4.3 (A*-26-A*-27).
handover for these use cases, careful planning of cell intersection sizes is required in order to guarantee large enough cell intersections for successful soft handover. Large intersections of cells imply employment of more network access points. Whether or not the increase in equipment cost and network topology planning is worth the complete independence of the security mechanisms has to be carefully decided.

In addition, some current wireless technologies (e.g., GSM, IEEE 802.11 [60, 92]) do not support soft handover procedures and consequently cannot support seamless service use if the establishment of a new security context via NAP_{DEST_k} is used.

### 3.2.2 Pre-Authentication Between MD and DEST_k via NAP\textsubscript{SRC_k}

A second method that uses a full authentication and key agreement is to run these protocols between MD and DEST\textsubscript{k} via NAP\textsubscript{SRC\textsubscript{k}}. In this case, SRC\textsubscript{k} forwards all traffic related to security suite negotiation, authentication, and key agreement between MD and DEST\textsubscript{k}, at some time before the handover execution. A pre-authentication like this one can be just in time, that is, right before HCN sends a handover request to DEST\textsubscript{k}. However, it can also take place pro-actively, before a handover reason is detected. A pro-active pre-authentication has the advantage that it does not add to the overall latency of the handover. In order to allow for pro-active pre-authentication, upcoming handover events have to be predicted and candidate destination networks have to be determined in advance. Although this problem is out of scope of this thesis, we will briefly discuss it in Section 6.4. Figure 3.11 exemplifies how this pre-establishment of a new security context can be integrated in a network-initiated handover procedure. Note that in this security solution, HN authorizes the handover by means of the initial authentication between MD and DEST\textsubscript{k}. However, the selection of DEST\textsubscript{k} and the initiation of the actual handover may be controlled by HN, AN, or DEST\textsubscript{k}.

In the mobile-initiated case, pre-authentication-based establishment of a new security context can be integrated into the procedure in the same way as on network-initiated handover. We, therefore, again do not detail the mobile-initiated case here.

The pre-authentication-based solution is limited by the following factors. First, SRC\textsubscript{k} has to forward all traffic related to security-suite negotiation and authentication and key agreement between DEST\textsubscript{k} and MD. This traffic has to be encapsulatable in messages exchanged between SRC\textsubscript{k} and MD. This can be difficult if the authentication and key-agreement protocols for a technology are implemented below a common network layer, i.e., as part of the MAC layer (see Section 1.2.2.7). The authentication and key-agreement protocols may then have to be adapted in order to support pre-authentication via SRC\textsubscript{k}.

Second, the execution of the authentication and key-agreement protocols have to be possible without causing the currently used data-protection keys to be replaced by the new ones immediately. Only this guarantees that the keys used before and after handover do not coincide.

Third, in case the pre-authentication is to be carried out just in time, the cell intersection has to be large enough to allow for a pre-authentication before MD leaves the cell. The required size of the intersection again depends on the velocity of MD, as well as the path
it takes through the intersection. The implications of this restriction have already been discussed in the last section.

Furthermore, carrying out the pre-authentication before a handover reason is detected may result in many unnecessary authentications that put an unnecessary load on both networks, as well as on MD. This problem has recently been addressed by integrating mechanisms that more precisely predict the NAP for the next handover using movement patterns of users (e.g., [138], [103], [102], [129]).

In summary, pre-authentication via SRC\textsubscript{k} is only applicable if the authentication and key agreement can be tunneled between MD and DEST\textsubscript{k} over NAP\textsubscript{SRC\textsubscript{k}}. Whether or not this is possible has to be determined for each technology. It is, however, important to note that in many current wireless technologies authentication and key agreement are implemented below the network layer, or at least involve the MAC layer which makes a tunneling over SRC\textsubscript{k} difficult. Consequently, pre-authentication via SRC\textsubscript{k} may be a viable solution as new technologies evolve, but cannot easily be used if current technologies, such as GSM, UMTS, CDMA2000, WLAN, or Bluetooth, are involved.
3.2 The Security Challenge and Solutions

3.2.3 Security-Context Transfer with Key Derivation

The advantage of the two solutions discussed so far is that the data-protection keys used to protect the data and control traffic before and after handover are derived from different master keys. Knowledge of the master key used before handover does not reveal any information on the master key used after handover and vice-versa.\textsuperscript{10} The master key used after handover is as strong as if it was established upon roaming to DEST\textsubscript{k}. The security suite to use are negotiated in the same way as on roaming. Consequently, HN has as much influence over each handover instance and on the choice of the security suite as the respective roaming procedures allow for.

As opposed to the two solutions discussed so far, in a solution based on \textit{SCT with key derivation}, MD and DEST\textsubscript{k} are not assured of each other’s authorization by a new, full run of a authentication protocol and do not agree upon a new master key \(K_k\) by means of a new run of a roaming key agreement. Instead, MD and DEST\textsubscript{k} are assured of each other’s authorization indirectly: HCN provides DEST\textsubscript{k} with a master key \(K_k\) derived from some previously used master session key by \textit{transferring} \(K_k\) to DEST\textsubscript{k} in a \textit{security context} during handover. MD derives this master key in the same way. By proof of possession of the same master key \(K_k\), MD and DEST\textsubscript{k} are assured of HCN’s authorization of the handover.

Note that in this chapter we do not further specify what information is included in the security context \(S_k\) transferred on a \(k\)-th order handover, except the fact that a master key \(K_k\) is part of the context.

SCT with key derivation is used to accelerate intra-provider handover in mobile phone networks (see, e.g., [60, 11]) and has recently been suggested, in the WLAN context as well [129, 185]. In other recent work [162, 75, 186, 27, 177], SCT with key derivation has been generalized to the inter-provider and even to the inter-system handover case. However, some of the above-mentioned work [186, 27, 129, 185] concentrates on a specific technology and none of the aforementioned work explicitly addresses the subsequent handover problem. We close this gap by explicitly generalizing SCT with key derivation to the three identified subsequent handover control types: HN-controlled, AN-controlled, and SRC-controlled handover. The relation between our SCT model and related work on SCT is further detailed in Section 3.3.2.

We start by modeling SCT with key derivation for first-order network-initiated handover with HN as anchor network and then proceed with subsequent handover including first-order and subsequent handover with FN as anchor, i.e., handover after roaming.

It is interesting to note that in the context transfer case, the differences between hard and soft handover do not play any role. In both cases, the security-context transfer is executed in exactly the same way. It is, therefore, sufficient to study only one type of procedure. We describe the hard handover cases, as SCT to date is the most promising solution to provide seamless handover in wireless technologies supporting only hard handover.

\textsuperscript{10}To be more precise, the master keys used before and after handover do not reveal any more information on each other than the master keys used on two subsequent roaming instances.
3.2.3.1 The Network-Initiated Case—First-Order with HN as Anchor

Prior to a first-order handover with HN as anchor network, MD and HN establish a connection as described in Figure 1.10. During the connection establishment, MD and HN negotiate the security suite \((a_0, ka_0, ke_0, em_0, im_0)\) to use, authenticate each other by means of the authentication protocol \(a_0\), and agree upon the master session key \(K_0\) by means of \(ka_0\). From \(K_0\) they derive and establish data-protection keys \(IK_0\) and \(EK_0\) by means of the negotiated key-establishment process \(ke_0\). They use the keys \(EK_0\) and \(IK_0\) as input to the encryption mechanism \(em_0\) and the integrity-protection mechanism \(im_0\) to secure the control and data traffic between MD and EIP\(_{HN}\).

The security suite and the master session key agreed upon between HN and MD is referred to as the initial security context throughout the remainder of this work.

**Definition 3.2.1** The initial security context \(S_0\) consists of the master session key \(K_0\) and the initial security suite \(ss_0\):

\[ S_0 = (K_0, a_0, ka_0, ke_0, em_0, im_0) =: ss_0 \]

For the security-context transfer with key derivation, MD and HN additionally have to negotiate a key-derivation function \(kd_0\). MD and HN can either fix \(kd_0\) during the pre-registration process or negotiate \(kd_0\) during connection establishment along with the security suite \(ss_0\).\(^\text{11}\) The key-derivation function \(kd_0\) takes the master key \(K_0\) and optionally other secret or public information as input. It outputs a master key \(K_1\) that, in the intra-system case we discuss here, is of the same length as \(K_0\). Upon handover, HN transfers a security context \(S_1\) to DEST that includes the derived master session key \(K_1\). We do not specify any further content of \(S_1\) at this point \(K_1\) is the main component of state-of-the-art SCT-based solutions.

Additionally, a cipher suite \(cs_1\) to use after handover must be negotiated. This cipher suite consists of a key-establishment process \(ke_1\), as well as encryption and integrity-protection mechanisms \(em_1\) and \(im_1\). The security mechanisms \(em_1\) and \(im_1\) are used in connection with keys \(IK_1\) and \(EK_1\) derived from \(K_1\) with the help of \(ke_1\).

As \(K_1\) is derived from \(K_0\), which has already been used to derive the data-protection keys \(EK_0\) and \(IK_0\), the security of the connection between HN and MD before handover and the security of the connection between MD and DEST after handover depend on each other. We will study the impact of SCT with key derivation in detail in the next chapter. It is, however, important to note at this point that the cipher suite negotiation upon handover should not only involve MD and DEST, but also HN.

We integrate four optional security-mechanism negotiation phases into our network-initiated handover procedure model. This phases will be used to integrate our new security-mechanism negotiation methods suggested in Chapter 5. In state-of-the-art handover solu-

\(^\text{11}\)In previously suggested SCT-based solutions (e.g., [129, 162]), MD and HN do not negotiate \(kd_0\) but agree upon a particular key-derivation function during pre-registration. We add the additional optional negotiation into our model to gain more flexibility.
tions MD and DEST\(_k\) negotiate the cipher suite to use after handover during association and without interaction with HCN or SRC\(_k\).

In the negotiation phase (1), HN and MD may exchange information about what security mechanisms they are willing to allow to be used after handover. In the negotiation phase (2), HN and DEST exchange policy information and in negotiation phase (3), MD and HN negotiate again dependent on the result of (2). Finally, in negotiation phase (4), MD and DEST negotiate the security mechanisms to use after handover. Note that these negotiation phases in the simplest case consist of single messages which can then be integrated with other messages, like the handover command, the handover request, or the handover response. In Section 5.1.1.5, we suggest several new security-mechanism negotiation methods.

Figure 3.12 describes a network-initiated first-order handover procedure with SCT of MD from HN to DEST. After detecting a handover reason and generating an ordered list

![diagram](image)

**Figure 3.12: First-Order Network-Initiated Handover with HN as Anchor Network**

L of candidate destination networks, HN sends a handover request to the first candidate destination network DEST\(^1\). The handover request includes the identities of MD, HN, and
DEST\textsuperscript{1}, as well as the security context $S_1$.

Upon receipt of the handover request, DEST\textsuperscript{1} decides whether to accept or refuse the handover request. During this decision process, DEST\textsuperscript{1} checks, e.g., whether it has the free capacity to serve MD or whether it is still in good standing with MD’s HN. DEST\textsuperscript{1} answers HN with a positive or negative handover response. If the handover response of DEST\textsuperscript{1} is positive, HN and DEST\textsuperscript{1} enter the security-mechanism negotiation phase(2). If the negotiation fails or the handover response of DEST\textsuperscript{1} was negative in the first place, HN restarts the selection process with the next candidate destination network DEST\textsuperscript{2} in $L$.

If the negotiation with a destination network DEST\textsuperscript{i} was successful, HN finally selects DEST\textsuperscript{i} as the destination network for the first-order handover (DEST\textsuperscript{i} = DEST) and sends a handover command to MD commanding handover to DEST. By sending the handover command, HN implicitly assures MD of its authorization of the handover.

If none of the candidate destination networks sends a positive handover response, or all negotiations fail, no inter-provider handover is possible and HN drops the connection with MD.

After receipt of a handover command, MD and HN may again negotiate on the cipher suite to use after handover. In case this negotiation fails, HN has to go back to the selection phase and again select the next destination network in its ordered list $L$.

If the negotiation is successful, MD disassociates from HN. MD and DEST associate with each other. They use the negotiated key-establishment process $ke_0$ to establish the data-protection keys $EK_1$ and $IK_1$ for the master session key $K_1$. Note that the possession of the same master session key $K_1$ indirectly assures MD and DEST of HN’s authorization of the handover. MD derives $K_1$ from $K_0$ by means of the key-derivation function $kd_0$, while DEST receives $K_1$ as part of the security context $S_1$ included in the handover request.

After successful association and key establishment, DEST sends the handover complete message to HN. Upon receipt of this message, HN releases the resources reserved for MD.

In the case of successful handover, MD and EIPE\textsubscript{DEST} use the negotiated mechanisms $em_1$ and $im_1$ to encrypt and integrity-protect data and control traffic between them.

### 3.2.3.2 The Network-Initiated Case—k-th-order Handover

A $k$-th-order handover procedure with SCT with key derivation, $(1 \leq k \leq h)$ is illustrated in Figure 3.13. This procedure model comprises first-order and subsequent handover with HN or FN as anchor. MD and AN establish connection at some time before the $k$-th-order handover procedure $(1 \leq k \leq h)$ and establish the initial security context

$$S_0 = ((r)a_0, (r)ka_0, ke_0, em_0, im_0).$$

Later, MD is subsequently handed over from AN = SRC\textsubscript{1} to DEST\textsubscript{1}, from DEST\textsubscript{1} = SRC\textsubscript{2} to DEST\textsubscript{2} and so on, until finally, MD is handed over to DEST\textsubscript{$k-1$} = SRC\textsubscript{k} on the ($k-1$)-st-order handover. Now, before the $k$-th handover, MD is connected to SRC\textsubscript{k}. SRC\textsubscript{k} and MD use the security mechanisms $em_{k-1}$ and $im_{k-1}$ to protect data and control traffic and derive the encryption key $EK_{k-1}$ and $IK_{k-1}$ from the master key $K_{k-1}$. If $k \leq 2$,
3.2 The Security Challenge and Solutions

SRC\(_k\) has received this master key \(K_{k-1}\) during the \((k-1)\)-st-order handover included in the security context \(S_{k-1}\). If \(k = 1\), SRC\(_1\) is AN and is in possession of \(K_0\) as a result of \((r)aK_0\).

If in this situation SRC\(_k\) detects a handover reason, it sends a handover indication message including the measurement data obtained from MD to HCN. HCN derives \(K_k\) and generates the security context \(S_k\). As will be detailed below, depending on the type of handover control, the master key \(K_k\) can be derived from \(K_0\) or from \(K_{k-1}\).

HCN sends a handover request including \(S_k\) to the first candidate destination network DEST\(_1^k\).

Upon receipt of the handover request, DEST\(_1^k\) decides whether to accept or refuse the handover request and sends its handover response back to HCN. In the case of a positive response, HCN and DEST\(_1^k\) enter the negotiation phase (2). In case the negotiation fails, HCN restarts the selection process and sends a handover request to DEST\(_2^k\).

If the negotiation with some DEST\(_i^k\) is positive, HCN selects DEST\(_i^k\) as destination network DEST\(_k\) and sends a handover command to SRC\(_k\) which forwards it to MD. The handover command includes the identity of DEST\(_k\). Optionally, MD and HCN now enter the negotiation phase (3).
After receipt of the handover command, MD disassociates from SRC\(_k\) and associates with DEST\(_k^i = \text{DEST}_k\). MD and DEST\(_k\) optionally enter the last negotiation phase (4) on the cipher suite to use after handover.

In the following sections, we briefly discuss the different handover control types.

**HN-Controlled Handover.** On HN-controlled subsequent handover, HN has full control over each handover instance. HN selects the next candidate destination network, and initiates and authorizes the actual handover. DEST\(_k\) receives a handover request from MD’s HN. DEST\(_k\) verifies the origin of a handover request as being a network with which it has a handover agreement. DEST\(_k\) is thereby assured of HN’s authorization of the handover instance. In case of commercial providers, HN reimburses DEST\(_k\) for the service provisioning to its pre-registered MDs.

HN includes \(K_k\) in the security-context transfer. HN may generate \(K_k\) either from \(K_{k-1}\) or from \(K_0\). In the first case, HN computes \(K_k = kd_0([K_{k-1}, []])\), where [] stands for other optional inputs we do not further specify here. We will discuss key-derivation functions in detail in Chapter 5. In the second case, \(K_k = kd_0(K_0, [])\). The former key derivation has the advantage that only one master key \(K_{k-1}\) at a time has to be stored by MD, as opposed to two keys in the second case (\(K_0\) for the derivation of future master keys and \(K_{k-1}\) as the currently used master key). However, the second key-derivation method has the advantage that depending on the properties of \(kd_0\), SRC\(_k\) may not gain any information on \(K_k\).\(^{12}\) If the first method is used by HN, SRC\(_k\) can derive \(K_k\) from \(K_{k-1}\) and consequently unnecessarily obtain \(K_k\). As will be detailed in Chapter 4, SRC\(_k\) could exploit this knowledge to impersonate MD to DEST\(_k\). We therefore assume that HN derives \(K_k\) from \(K_0\) on HN-controlled handover.

In order to be able to derive the sequence of master keys, HN has to get into the possession of the initial master key \(K_0\). In case HN is the anchor network of the handover chain, HN gets knowledge of the initial master key \(K_0\) during the initial (roaming) authentication and key agreement (see Chapter 2). If the anchor network is a foreign network, then whether or not HN gets to know \(K_0\) during the initial authentication depends on the chosen roaming key-agreement protocol. If HN does not get to know \(K_0\) during authentication, FN has to transfer \(K_0\) along with the handover indication to HN over a secure channel on the first-order handover from FN to DEST\(^1\).

**AN-Controlled Handover.** AN-controlled handover can be implemented in almost the same way as HN-controlled handover. The main difference in the AN-controlled case lies in the fact that the next destination network DEST\(_k\) of a \(k\)-th-order handover is selected by AN, which in turn may be HN or FN, and it is AN that initiates and authorizes the handover. In case of commercial networks, HN reimburses AN and AN reimburses all subsequently serving networks. AN derives the master key \(K_k\) and transfers it in the handover request to DEST\(_k\). AN may derive \(K_k\) from \(K_0\) or from \(K_{k-1}\). With the argument given above,

\(^{12}\)We will show in Section 6.1 that this has an additional advantage on inter-system handover, as different technologies typically require different master key lengths.
we assume AN derives $K_k$ from $K_0$. As opposed to HN, AN gets to know $K_0$ during any type of initial roaming authentication protocol, such that no additional transfer of $K_0$ is necessary.

**SRC-Controlled Handover.** On SRC-controlled subsequent handover, HN delegates the handover decision and grants authorization to AN during authentication. AN subsequently delegates the handover control and authorization to DEST$_1$ = SRC$_2$ and so on. Handover is possible only if SRC$_k$ and DEST$_k$ have a handover agreement. This agreement is not only valid for their own respective pre-registered users, but also for other MDs that are currently connected to SRC$_k$. In case of commercial networks, HN reimburses AN, AN reimburses DEST$_1$, DEST$_1$ reimburses DEST$_2$ and so on for service provisioning to MD.

On a $k$-th-order handover, SRC$_k$ sends a handover request to DEST$_k$ and DEST$_k$ verifies that it originates from a SRC$_k$ with which it has a handover agreement. If this is the case, DEST$_k$ assumes the handover request as being authorized by SRC$_k$. SRC$_k$ includes the security context with the master key $K_k$ in the handover request.

SRC$_k$ derives the master key $K_k$ from the previously used master key $K_{k-1}$ with the help of a key-derivation function $kd_{k-1}$. SRC$_k$ and MD negotiate $kd_{k-1}$ as part of the cipher-suite negotiation on the $(k-1)$-st-order handover.

An obvious disadvantage of this key-derivation method is that SRC$_i$ ($0 \leq i \leq k - 1$) gains knowledge of the master keys $K_{i+1}, \ldots, K_k$ if no additional secret input parameters to the key-derivation functions are used. Each subsequently serving network consequently not only has to trust its predecessor, but *all* previously serving networks with which they possibly do not have any handover agreement or any other type of prior trust relationship. The same holds true for MD, as it is assured of AN’s authorization by HN to offer service to MD during the initial authentication, but has to transitorily trust all of the subsequently serving networks.

An HN-assisted (or AN-assisted), SRC-controlled subsequent handover could be used to thwart this threat. HN (or AN) sends a list of allowed handover destination networks to MD during authentication. Upon subsequent SRC-controlled handover, MD verifies that DEST$_i$ is included in the list of allowed destination networks. This ensures that handover only take place if HN and SRC$_k$, as well as HN and DEST$_k$ and SRC$_k$ and DEST$_k$, have handover agreements. As opposed to this on purely SRC-controlled handover, HN and DEST$_k$ do not necessarily have to have a handover agreement.

In any type of handover control, SCT with key derivation makes the protection of the connection before handover dependent on the protection of the connection after handover and vice-versa. For a handover procedure within the network of one provider, a key-derivation approach is acceptable as long as the EIPEs of the network support the same security mechanisms.¹³ For inter-provider handover procedures, however, SCT with key derivation calls for answers to the following questions:

---

¹³If the EIPEs of a network support different security mechanisms, the intra-provider case is identical with inter-system handover within the same provider. This will be shown and explained in more detail in Section 6.1.
1. To what extent can DEST\(_k\) trust the AN's authentication of MD?

2. To what extent can DEST\(_k\) trust the intractability of the master key received from HCN?

3. To what extent can MD trust the intractability of the derived master key?

4. How should DEST\(_k\) and MD negotiate the cipher suite to use after handover and how can the policies of previously serving networks be respected during this negotiation?

Although SCT with key derivation has recently been extensively studied (e.g., [186, 177, 74, 185, 129, 55, 75, 162]), satisfying answers to the above questions are not easily found. The threats arising from the dependency of the keys have not been adequately treated in literature so far and are often underestimated (see [75, 111]). We will close this gap in Chapter 4 and give an extensive threat analysis for SCT with key derivation.

SCT with key derivation does not add to the disruption time on hard handover. Consequently, SCT-based solutions are the method of choice to support seamless handover procedures in case the two aforementioned methods cannot be applied. In particular, SCT with key derivation has to be used if seamless hard inter-provider handover is to be offered, or high velocity of MDs has to be expected.

3.2.3.3 Differences in the Mobile-Initiated Case

SCT with key derivation is very similar for mobile-initiated and network-initiated handover. For all types of handover control, HCN generates the security context on mobile-initiated handover in the same way as on network-initiated handover. The two main differences are that the negotiation of the cipher suite to use after handover takes place at different points in the procedure and that in case HCN is notified by DEST\(_k\) of the mobile-initiated handover, DEST\(_k\) pulls the security context from HCN, while in case HCN is notified by MD, HCN pushes the security context to DEST\(_k\).

As in the network-initiated case, the security-context transfer on hard and soft handover is very similar. We therefore describe hard handover procedures only.

**HCN Notified by DEST\(_k\).** The security-context transfer for a \(k\)-th-order mobile-initiated handover where HCN is notified by DEST\(_k\) is illustrated in Figure 3.14.

As opposed to the network-initiated case, HCN transfers the security context \(S_k\) to DEST\(_k\) in the handover request following HCN's handover decision. HCN is assured by the handover indication that MD in fact indicated handover to DEST\(_k\).

In Figure 3.14, we identify three phases in which MD, DEST\(_k\), and HCN can negotiate the cipher suite to be used after handover. In phase (1), MD and DEST\(_k\) negotiate as part of the association process. In phase (2), HCN and DEST\(_k\) negotiate and in phase (3), MD and DEST\(_k\) may again negotiate again depending on the results of the previous negotiation phases.
3.2 The Security Challenge and Solutions

HCN Notified by MD. Security-context transfer for a $k$-th-order mobile-initiated handover where HCN is notified by MD is illustrated in Figure 3.15.

HCN is notified by MD with the help of SRC$_k$’s forwarding of the handover indication. HCN transfers the security context $S_h$ in the handover request to DEST$_k$.

We identify three possible phases during the procedure in which MD, DEST$_k$, SRC$_k$, and HCN can negotiate the cipher suite to use after handover. In phase (1), MD and SRC$_k$ can negotiate before a handover reason is detected. In phase (2), HCN and DEST$_k$ can negotiate right after HCN sends the handover request to DEST$_k$. In phase (3), MD and DEST$_k$ can negotiate after or as part of DEST$_k$’s handover response.

3.2.3.4 Post-Authentication After SCT with Key Derivation

As we will detail in Chapter 4, SCT with key derivation brings up various threats for MD SRC$_k$, DEST$_k$, HN and AN. One way to reduce the risk taken by related master keys is using a post-authentication after handover. The purpose of this post-authentication is to mutually authenticate MD and DEST$_k$ based on an authentication protocol and agree upon new keys between DEST$_k$ and MD by means of a roaming key agreement as soon as the
handover is successfully completed. A post-authentication is possible only if the destination technology supports a change of data-protection keys during an ongoing service use.

In case post-authentication is used, the master session key transferred on a $k$-th-order handover only depends on the master key newly generated between MD and SRC$_k$ after the $(k - 1)$-st-order handover. Consequently, $k$-th-order handover with SCT with key derivation and post-authentication do not differ from first-order handover of the same type.\textsuperscript{14}

Whenever SCT with key derivation is used, a post-authentication should at least take place as soon as MD is in the so-called \textit{idle} or \textit{dormant mode} (see [116] for precise definition), i.e., as soon as soon as MD is still associated with the source network but no longer uses any service.

\textsuperscript{14}In the case of HN-controlled handover, an authentication mechanism by which HN does not get to know the master session key and SRC$_k$ has to send the newly generated key to HN along with the handover indication. In the case of AN-controlled handover, SRC$_k$ has to transfer the newly generated key to AN along with the handover indication. Consequently, post-authentication seems to be most suitable in connection with SRC-controlled or HN-assisted handover.
3.2.4 SCT with Key Agreement

The first two security solutions presented are based on a full roaming authentication and key agreement and thus have the advantage that new independent keys are used after handover. However, SCT with key derivation is clearly more efficient than these solutions, as it requires fewer round-trip message exchanges between MD, DEST_k, and, depending on the authentication type, HN. This efficiency advantage makes SCT attractive for all types of handover procedures. A mixed method combines the advantages of newly agreed-upon keys with the efficiency of a context transfer: a new master key $K_k$ is agreed upon based on the credentials exchanged between MD and HN. The HCN then transfers a security context $S_k$ to DEST_k upon handover that includes public or secret information on $K_k$ (i.e., $K_k$ itself, some random number by means of which DEST_k can compute $K_k$, or the like).

The message exchange on a handover using SCT with key agreement is the same as in the case of SCT with key derivation. The two differences are the way the master session key $K_k$ is generated and the fact that the handover participants have to negotiate a key-agreement method (rather than a key-derivation method) to use to agree upon $K_k$.

In this section we briefly introduce two key-agreement methods that generalize state-of-the-art intra-provider handover security solutions.

In the first key-agreement method, MD and SRC_k run a key-agreement protocol before a handover reason is detected. By means of this key-agreement protocol, MD and SRC_k agree upon a new master session key $K_k$ in the same way as if MD was roaming to SRC_k. Instead of using this master key right away, $K_k$ is transferred to DEST_k upon handover. For this purpose, HCN has to obtain possession of $K_k$.

In the HN-controlled case, HN generates the security context $S_k$ and includes the fresh key $K_k$ in $S_k$. HN gets knowledge of the new key in one of two ways: either HN is involved in the roaming key-agreement protocol, as it goes back to HN anyway or because HN is the anchor network of the handover, or SRC_k sends the fresh master key $K_k$ in the handover request to HN.

In an AN-controlled handover with HN as anchor, the same argument holds true. In an AN-controlled handover with FN as anchor, SRC_k has to provide AN with the newly generated master key $K_k$.

In an AN-controlled handover, SRC_k is in possession of the master key $K_k$ anyway. SRC-controlled handover is the control type of choice for this variant of SCT with key agreement, as it minimizes the number of key transfers necessary.

Figure 3.16 illustrates how the new roaming key agreement $(r)ka_k$ can take place between MD and SRC_k before a handover is even detected.

The new key agreement takes place between SRC_k and MD at any time before the handover execution and possibly even before the detection of a handover reason. As a user may start to use a service immediately after the initial authentication, the additional key agreement may have to be executed during an ongoing service use. This restricts the use of

---

15In case authentication and key agreement are implemented together, a pair $(r)a_k, (r)ka_k$ of authentication and key-agreement protocols has to be negotiated.
SCT with this key agreement to technologies that support roaming key agreement during an ongoing use of service.

To circumvent this restriction, we introduce a second example for SCT with key agreement. The initial roaming key-agreement protocol can be changed such that several independent keys are generated at a time. The number of subsequent handover is then restricted to the number of independent keys generated. A counter has to keep track of the number of subsequent handover. It is important to note that this variant differs from the first key-agreement method described above in that all subsequently used master keys are generated by the AN or by HN, depending on where the master key is generated upon roaming. In case HN does not generate the master session keys and an HN-controlled handover is to be implemented, the anchor network, if different from HN, has to transfer all master session keys to HN in the first handover request. In the case of a SRC-controlled handover, each source network $SRC_k$ has to transfer all yet unused master keys to DEST$_k$ in the security context of the $k$-th-order handover. Consequently, AN-control seems to be the best handover control type for this version of SCT with key agreement.

Figure 3.17 illustrates the generation of multiple master keys. The handover procedure itself, including the security-mechanism negotiation, is the same as in the case of SCT with key derivation. The difference lies only in the way the subsequently used master keys are generated.

---

Figure 3.17 comprises network-initiated and mobile-initiated handover procedures.
3.2 The Security Challenge and Solutions

Using SCT with key derivation on a mobile-initiated handover differs in the same way from SCT with key derivation as in the network-initiated case. We therefore refrain from detailing the mobile-initiated case.

The data-protection keys used after handover are derived from freshly generated master keys. An attacker in possession of the keys used before handover can therefore not use this knowledge to break the keys used after handover. The connection between MD and DEST$_k$ after handover is as well-protected as a newly established connection between MD and SRC$_k$. The roaming key-agreement used may, however, be a protocol that DEST$_k$ does not support itself. Consequently, the connection between MD and DEST$_k$ may not be protected in the same way as a newly established connection between DEST$_k$ and MD. It is also important to note that although independent keys are used before and after handover, in our procedure, the master key used after handover becomes known to SRC$_k$, as well as HCN. In order to avoid this in the first-order handover case, Wang et al. in [177] suggest using the transferred master key to authenticate a Diffie-Hellman key exchange [53] between MD and DEST$_k$ after handover. The key generated by the Diffie-Hellman exchange is then used to derive the data-protection keys and not the transferred master key. However, the authors fail to notice that by means of its knowledge of the master key SRC$_k$ can mount a man-in-the-middle attack against the key exchange, thus determining the master key used between MD and DEST$_k$. Wang et al.’s solution requires SRC$_k$ to perform an active attack in order to obtain the master key, but it does not protect against disclosure of the master and data-protection keys to SRC$_k$.

Any SCT with key agreement calls for answers to the following questions:

1. To what extent can DEST$_k$ trust the AN’s authentication of MD?

Figure 3.17: Multiple Initial Key Generation for SCT with Key Agreement
2. To what extent can DEST\textsubscript{k} trust the intractability of $K_k$?

3. How and when should MD and DEST\textsubscript{k} negotiate the cipher suite (and possibly also the key-agreement protocol $ka_k$) to use after handover and to what extent should the previously serving networks and HCN be involved in the negotiation?

We will discuss the threats arising from SCT with key agreement in Section 4.4.

3.3 Related Work

3.3.1 Handover Procedures

In [184], Zdarsky et al. give an overview on existing handover procedures and current trends. Moreover, the authors compare the advantages and disadvantages of mobile-initiated and network-initiated handover procedures. However, security issues are not addressed.

Zhang et al. [187] give an overview on the criteria by which in current wireless access technologies handover reasons are detected and candidate destination NAPs are determined.

Network-initiated handover procedures are specified for GSM, UMTS, and CDMA2000 in [60, 11, 13]. Although these procedures are specified for intra-provider handover, our generalization to the inter-provider handover case is straightforward and also complies with the procedure described in [17].

Handover between different WLANs are currently standardized as part of 802.11f [92]. The procedure currently followed by most WLAN cards is described by Mishra et al. in [127]. This procedure corresponds to the mobile-initiated procedure we have modeled, in which HCN is notified by DEST\textsubscript{k}. It is, however, important to note that in [127] only intra-provider handover are considered.

Mobile-initiated handover procedures where HCN is notified by DEST\textsubscript{k} are also described by Wang et al. [176] (requested transfer scheme signaling), Xhafa et al. [182], and Oyoqui et al. [135]. The authors of these papers consider first-order handover after roaming, but they do not address subsequent handover.

Mobile-initiated handover where HCN is notified by MD are, for example, specified in Wang et al. in [176] and referred to as active transfer scheme signaling. Again, the authors only address handover after roaming, but not subsequent handover.

3.3.2 Security Solutions

The pre-authentication method and the SCT with key agreement we have presented in this chapter are generalizations of the pre-authentication method used in 802.11i [93], as well as the predictive authentication method suggested by [136]. The pre-authentication method of [93] corresponds to executing a new run of the key-agreement protocol while still connected to SRC\textsubscript{k}. In [93], only intra-provider handover are considered. We generalize this approach to inter-provider handover in two different ways: the pre-authentication with DEST\textsubscript{k} over SRC\textsubscript{k} and the SCT with key agreement by means of a new run of the key-agreement protocol between SRC\textsubscript{k} and MD.
3.4 Conclusion

The predictive authentication method presented by Pack et al. in [136] is generalized to generating multiple keys during the initial roaming key agreement with AN. It is interesting to note that in order to accelerate the handover execution, Pack et al. suggest distributing the multiple master keys to the destination access point before a handover reason is detected, rather than using context transfer during handover. To select the candidate network access points, they suggest an algorithm to determine a Frequent Handover Region (FHR).

On intra-provider handover within the mobile telecommunication technologies GSM, UMTS, and CDMA2000, the data-protection keys coincide with the master key generated during the roaming key agreement [60, 11, 13]. Upon handover, these data-protection keys are, without any modification, transferred to the next network access point (i.e., $ke = id$ and $kd = id$). The cipher suite used may, however, change. The impact of this SCT strategy is discussed in detail in Chapter 9.

Security-context transfer has recently been discussed for inter-provider and even inter-system handover [186, 177, 74, 185, 75, 162]. While [75, 74, 176] concentrate on how SCT could be integrated and used to accelerate handover and on defining security requirements for SCT, some of this work [177, 162, 185, 186] makes concrete suggestions on how the security context could be generated and what information it should include. However, none of the work explicitly addresses subsequent handover. We will discuss the above SCT-based solutions in more detail in the Related Work section at the end of the next chapter.

SCT is also under research in the SEAMOBY working group of the IETF, where requirements and contents of the transfer are still under discussion [79]. The main result of this group so far is the Context Transfer Protocol [111], which specifies context transfer between two access routers whenever a user’s mobility makes a quick re-establishment of ongoing sessions necessary. However, the protocol focuses on the actual transfer messages rather than on their content. In particular, suggestions for the content of security-related context information are not included in [111].

3.4 Conclusion

In this chapter, we have introduced a new formal model for various types of inter-provider handover procedures. As opposed to prior work, we have distinguished between first-order and subsequent handover procedures and have identified and described three new handover control types. We have modeled the security challenges arising from inter-provider handover and have presented four different approaches to address these challenges. These approaches generalize the solutions that are currently used to secure intra-provider handover or were previously suggested for first-order inter-provider handover to our newly modeled inter-provider subsequent handover procedures.

An interesting topic for future research would be to develop new accounting schemes and business models for subsequent inter-provider handover for each handover control type defined in our new formal model.
Chapter 4

Threat Analysis for Security-Context Transfer on Inter-Provider Handover within the Same Technology

The impact of SCT on handover across administrative boundaries is not very well understood yet [75, 111]. SCT has widely been identified as the most promising security solution to meet the efficiency requirements imposed by inter-provider handover procedures (e.g., [186, 177, 74, 185, 129, 55, 75, 162]). However, much of the previous work on this subject does not address the impact of SCT on the security goals of users and providers and does not specify any security requirement for SCT. The need for a thorough threat analysis for SCT on handover was previously stated in [75, 111], and first steps to specify security requirements were taken in [162, 176, 177]. However, none of the previous work explicitly addresses the problems arising from subsequent inter-provider handover. In particular, subsequent context transfers with key derivation make the protection of the connection after handover dependent on the protection of each connection between MD and a previously serving network. Moreover, previous work fails to meet the interest of previously serving networks in the negotiation of the cipher suite to use after handover. Instead, it is implicitly assumed that all network providers offer the same level of protection and that MD and DEST will try to maintain the protection level upon handover. This assumption, however, is unrealistic, as different providers will typically support different security mechanisms and, depending on the type of handover agreement, different providers will take different risks if weak mechanisms are used after handover.

In this chapter, we present a threat analysis of the two different SCT types modeled in Section 3.2.3 and Section 3.2.4 of the last chapter. From this analysis, we derive new security requirements for both forms of SCT. In Chapter 5, we present new handover procedures that use SCT and meet (most of) the newly defined requirements.
We begin our threat analysis by identifying potential attack goals of an attacker, namely violating the confidentiality of the air interface between MD and a wireless access network, violating the integrity protection between MD and a wireless access network, mounting Denial of Service (DoS) attacks against MDs or networks, and conducting service theft against a network on behalf of a victim MD. We describe different attack scenarios specifying the context in which an attacker could try to achieve the above attack goals. Finally, we describe several attacks for each attack scenario.

In particular, we show how an attacker can try to exploit weaknesses in the message exchange on handover execution in order to mount impersonation and DoS attacks against MDs and networks. Furthermore, an attacker can exploit weaknesses in the detection of handover reasons and the selection of candidate destination networks in order to mount DoS and impersonation attacks. In addition, we show how an attacker can exploit weaknesses introduced by the key relations (SCT with key derivation) in combination with weaknesses in the initial security suite or the cipher suite used between MD and a previously serving network in order to mount attacks against the confidentiality and integrity protection or to mount service theft attacks. We also show how an attacker can exploit bidding down attacks against the initial security-suite negotiation, as well as against the cipher-suite negotiation on a previous handover, in order to mount various attacks.

In our threat analysis we use two equivalent ways to describe the attacks identified for each attack scenario. One of them makes use of so-called attack trees [158]. Attack trees provide a formal but very intuitive method of describing the security of a system based on varying attacks starting from root attack scenario down to the initial steps an attacker has to achieve in order to mount an attack in the root scenario. The second way to describe attacks identifies recurring modules an attacker can combine in order to mount more sophisticated attacks. Once recurring attack modules are identified and all attacks are described with their help, protection mechanisms can more easily be evaluated. If each attack requires the use of at least one attack module and the protection mechanisms applied protect against all attack modules, the system is secure against all identified attack. However, identifying attack modules from scratch is not an easy task. It is by far easier and more intuitive to first construct attack trees, then identify recurring attack modules and then design protection mechanisms that protect against the identified attack modules, thus making use of the advantages of each description method.

We use attack trees in order to describe potential attacks against a first-order handover with HN as anchor for each attack scenario. We then identify recurring attack modules (subtrees) in these trees and described each identified attack with the help of these attack modules. We then generalize the modules identified for first-order handover with HN as anchor to modules for k-th-order handover (k \( \geq 1 \)) with FN or HN as anchor and describe attacks against a general k-th order procedure with the help of the generalized modules. We show that more sophisticated impersonation attacks can be mounted against subsequent handover. This is due to the fact that an attacker cannot only try to impersonate MD to the anchor network and the destination network (and vice-versa), but he can also try to exploit the handover procedure in order to impersonate MD to any of the previously serving
From the threat analysis, we derive requirements and make recommendations to enhance SCT on inter-provider handover. We show that a handover procedure that meets these requirements is secure against most of the identified attack modules and, consequently, against most of the attacks.\footnote{Our requirements do not address protection against DoS attacks and attacks that are solely based on compromising the memory or communication between network components within a network.} In Chapter 5, we present our new history-enriched, policy-based handover procedures that meet the newly defined requirements.

**Outline.** In Section 4.1, we introduce our notations for attack trees. In Section 4.2, we analyze the threats arising from first-order inter-provider handover with HN as anchor in the case that SCT is used with key derivation. This threat analysis is generalized to the $k$-th order handover case with an arbitrary anchor network in Section 4.3. We describe the differences on SCT with key agreement in Section 4.4. In Section 4.5, we detail how our contributions in this chapter relate to previous work in the field. Finally, we conclude with a summary of the contributions of this chapter in Section 4.6.

### 4.1 Attack Trees

Attack trees provide a formal method of describing the security of systems based on varying attacks. Attacks against a system are represented in a tree structure. The root of an attack tree describes the goal the attacker wants to achieve in a certain scenario. The children of a node represent different ways to achieve the parent node. Each node in the tree thus becomes a subgoal for an attacker that wants to achieve the goal in the scenario at the root of the tree. There are AND and OR nodes. OR nodes represent different alternatives to achieve the parent node. AND nodes represent different steps that have to be taken in combination to achieve the parent node. AND and OR nodes can appear as children of the same parent node. Moreover, several pairs (or larger collections) of AND nodes can appear as children of the same parent node. Attack trees are described in [158].

Read from the bottom to the top, each branch in an attack tree that contains only OR nodes can be interpreted as a sequence of steps an attacker has to perform in order to achieve the root goal in the attack scenario. Branches that contain AND nodes have to be bundled together to form a complete attack achieving the root goal in the scenario.

Figures 4.1 to 4.6 summarize the notations used for attack trees here. Root attack scenarios are represented by squares with cut-off corners. Regular subgoals, that is subgoals that do not reappear in other trees, are represented by boxes. Subgoals that reappear in other trees (reappearing subgoals) are represented by ellipses. In order to enhance readability, the subtrees starting at these subgoals are described in separate trees. Only subtrees that are independent of all other subtrees within an attack tree can be described separately. Otherwise, extracting a subtree destroys the structure of the attack tree. In Figure 4.4, the nodes B and C are alternatives to achieve the (sub)goal A. B and C are represented as children of A connected to A by solid lines. This is the notation used for OR nodes here.
In Figure 4.5, the nodes B and C are different steps that have to be taken in combination to achieve A. B and C are again children of A, and they are connected to A by dashed lines or lines. We use this notation for B and C are AND nodes here. Finally, Figure 4.6 demonstrates how AND and OR nodes can lead to one and the same subgoal. In this figure, A can either be achieved by B and C in combination or by D alone.

We cannot prove that any of our attack trees is complete. However, the use of attack trees makes it easy to check whether a given attack was considered or not and additional attacks can easily be added into a tree over time.

4.2 First-Order Handover with HN as Anchor and SCT with Key Derivation

4.2.1 Root Attack Scenarios

In this section, we describe the root attack scenarios we consider for an inter-provider first-order handover with HN as anchor. These attack scenarios themselves are independent of the initiation type of the handover procedure, in other words, they are the same for network-initiated and mobile-initiated handover. However, the attack trees for each root scenario differ for network-initiated and mobile-initiated handover procedures.

In the first five scenarios the goal of an attacker is to violate the confidentiality of user and control traffic sent or received by a victim MD.

In the first scenario, an attacker intercepts and records encrypted user or control traffic on the air interface between MD and HN before a handover takes place. Some time later, MD is handed over from HN to an authorized destination network DEST. The attacker tries to exploit this handover procedure to gain access to the plaintext of previously recorded traffic. This Root Attack Scenario (RAS) can be summarized as:
RAS-1 An attacker recovers the plaintext of encrypted data or control traffic he intercepts and records on the air interface between MD and HN after a handover of MD from HN to an authorized DEST takes place.

The second scenario is restricted to hard handover procedures. A MD is about to be handed over from HN to some destination network DEST. HN sends the handover command to MD, and MD disassociates from HN and tries to associate to DEST. An attacker tries to prevent the handover of MD by simulating a handover failure, impersonates HN to MD when MD tries to re-associate with (full back to) HN, and then tries to recover the plaintext of data or control traffic sent by MD:

RAS-2 An attacker recovers the plaintext of data or control traffic sent by MD by impersonating HN to MD on a simulated handover failure.

In the third scenario, MD is handed over from HN to an authorized destination network DEST. An attacker intercepts data or control traffic on the air interface between MD and DEST after handover (as opposed to intercepting the traffic between MD and HN before handover in RAS-1). He tries to exploit the handover procedure to recover the plaintext of the intercepted encrypted traffic after handover.

RAS-3 An attacker recovers the plaintext of data or control traffic exchanged between MD and an authorized DEST after a handover of MD from HN to DEST takes place.

In addition, we study two other scenarios in which also the recovery of plaintext of confidential traffic is the goal of the attacker. Unlike in RAS-2, where the attacker impersonates a NAP of HN, here the attacker tries to recover the plaintext of data or control traffic of an ongoing connection by impersonating a destination NAP to MD on a handover procedure.

RAS-4 An attacker gains access to the plaintext of data or control traffic sent by MD by impersonating NAPDEST on an actual handover of MD from HN to DEST.

RAS-5 An attacker gains access to the plaintext of data or control traffic sent by MD by simulating a handover from HN to DEST and impersonating NAPDEST to MD.

In the scenarios described so far, the goal of the attacker is to violate the confidentiality of data or control traffic. In the next scenario he aims to violate the integrity of data or control traffic. The root attack scenario we consider here is the following: after or while MD is handed over from HN to an authorized DEST, an attacker tries to manipulate data or control traffic exchanged between MD and DEST.

RAS-6 An attacker manipulates data or control traffic between MD and an authorized DEST after or during handover.
Note that an attacker may be able to decrypt data or signaling traffic exchanged between MD and HN before handover due to knowledge he gained during or after handover (see RAS-1). However, an attacker cannot use any knowledge gained during or after handover in order to manipulate traffic exchanged between HN and MD before handover.

Another possible goal of an attacker is service theft. An attacker tries to gain access to the network of HN or DEST and to use services on behalf of a victim MD. In commercial networks, the victim MD ends up paying for the service used by the attacker. In non-commercial networks, using a service on behalf of MD may result in unauthorized access to confidential data and the ability to cause damage to accessible resources without being traceable. We consider the following two root attack scenarios in which service theft is the goal of an attack:

**RAS-7** An attacker tries to gain access to HN’s network on behalf of a victim MD exploiting an actual handover procedure (service theft against HN).

**RAS-8** An attacker tries to gain access to DEST’s network on behalf of a victim MD on an actual handover of MD from HN to DEST (service theft against DEST).

Wireless networks are particularly vulnerable to DoS attacks. An attacker can easily jam MDs or network access points. It can send out false requests and other traffic to keep the network access points and other network components busy. We consider three DoS scenarios here, one for each of the participants of a first-order handover procedure with HN as anchor: MD, HN, and DEST.

**RAS-9** An attacker prevents a legitimate MD from continuously using HN’s or DEST’s service by interfering with the handover procedure (denial of service attack against MD).

**RAS-10** An attacker uses the handover procedure to block resources, like a channel allocated for MD, in HN (denial of service attack against HN).

**RAS-11** An attacker uses the handover procedure to overload an authorized DEST (denial of service attack against DEST).

Note that attacks based on stolen or cloned MDs are not considered here. Stolen or cloned MDs have to be ruled out during the initial authentication between MD and AN. Once they are connected to a network, they can be handed over from one network to another, as they are assumed to be authorized by the handover controlling network due to the initial authentication.

### 4.2.2 Attack Tree for Root Attack Scenario RAS-1

In this section, an attack tree for the first of the identified root attack scenarios RAS-1 is described and discussed for a network-initiated first-order handover procedure. The purpose
4.2 First-Order—HN as Anchor—SCT with Key Derivation

of this example tree is to motivate how we obtained the attack modules and attacks described in the next sections in a methodical way. Attack trees for the other root attack scenarios can be found in Appendix A.

It is important to note that the leaves in our attack trees themselves are attack modules against the network components or protection mechanisms used in between MD and HN before handover or between MD and DEST after handover. The attack tree notation thus provides means to determine how far an attacker that can mount one or more of these attack modules can get in achieving the attack goal in the root scenario. Attack trees thus allow for a “what if” argumentation. Whether or not an attacker can mount the attack modules described as the leaves of the attack trees has to be analyzed for each wireless access technology separately.

The full tree for the root attack scenario RAS-1 does not fit on one page and is therefore split into several smaller figures, namely Figures 4.7 to 4.12. Splitting the tree into several subtrees has the additional advantage that subgoals reappearing in more than one attack tree can be described once and can then easily be referred to.

Figure 4.7 reads from the top to the bottom as follows: data or control traffic is encrypted with the encryption mechanism \( em_0 \) and the encryption key \( EK_0 \). In order to recover the plaintext of encrypted traffic, the attacker either has to find some means to recover the encryption key \( EK_0 \) or has to be able to recover the plaintext without the encryption key. In compliance with [120], the second alternative is referred to as partially break \( em_0 \).
Chapter 4. Threat Analysis for Security-Context Transfer

Recovers $E_{K_0}$

Partially breaks $e_{M_0}$

Recovers $E_{K_0}$ with knowledge of $K_0$

Recovers $E_{K_0}$ without knowledge of $K_0$

Recovers $K_0$ without knowledge of $E_{K_0}$, $I_{K_0}$

Reconstructs $k_{e_0}$

Recovers $K_0$ without knowledge of $E_{K_0}$, $I_{K_0}$, and $K_1$

Recovers $K_0$ without knowledge of $E_{K_0}$, $I_{K_0}$, but with knowledge of $K_1$

Recover $K_1$ without knowledge of $K_0$.

Recovers $K_1$ without knowledge of $E_{K_1}$, $I_{K_1}$, and $K_0$

Recovers $K_1$ without knowledge of $K_0$ but with knowledge of $E_{K_1}$, $I_{K_1}$

Recovers $E_{K_1}$ without knowledge of $K_1$

Recovers $I_{K_1}$ without knowledge of $K_1$

Totally breaks $e_{M_1}$ without knowledge of $E_{K_0}$

Compromises EIPE_DEST and reads $E_{K_1}$ from memory

Compromises EIPE_DEST and reads $I_{K_1}$ from memory

Totally breaks $e_{M_1}$ without disabling $e_{M_0}$

Compromises EIPE_DEST and reads $K_1$ from memory

Totally breaks $e_{M_1}$ without disabling $e_{M_0}$

Compromises EIPE_DEST and reads $K_1$ from memory

Attacker recovers plaintext of encrypted traffic intercepted and recorded on the air interface between MD and HN after a handover of MD from HN to DEST takes place.

Figure 4.7: Attack Tree for Root Attack Scenario RAS-1
Definition 4.2.1 An attacker can partially break the encryption mechanism \( \text{em} \) if and only if he can recover the plaintext of data encrypted with \( \text{em} \) without being able to recover the encryption key.

The attacker can recover \( EK_0 \) with or without the knowledge of \( K_0 \). How the attacker can recover \( EK_0 \) without knowledge of \( K_0 \) is illustrated in Figure 4.8. The attacker can recover \( EK_0 \) without knowledge of \( K_0 \) by totally breaking \( \text{em}_0 \).

Figure 4.8: Attack Tree for Subgoal “Recover \( EK_0 \) without Knowledge of \( K_0 \)”

\( EK_0 \) without knowledge of \( K_0 \) by totally breaking \( \text{em}_0 \).

Definition 4.2.2 An attacker can totally break the encryption mechanism \( \text{em} \) if and only if he can recover the encryption key \( EK \) by cryptoanalyzing \( \text{em} \).

An example for cryptanalysis of an encryption mechanism is a ciphertext-only attack by which the attacker can recover the encryption key from the intercepted and recorded ciphertext. Other examples for cryptanalysis are known-plaintext attacks, or chosen-ciphertext attacks. While a ciphertext-only attack can be mounted by an attacker by simply intercepting and recording encrypted traffic on the air interface, other cryptanalysis attacks require the attacker to take additional steps to mount an attack. How and whether these attacks succeed highly depends on the actual encryption mechanism used and, consequently, cannot be studied on the abstraction level of this model. A cryptanalysis is generally easier the more traffic encrypted with the same key can be obtained. The lifetime of an encryption key should therefore be restricted.

The attacker can try to break whatever encryption mechanism \( \text{em}_0 \) HN and MD agree upon. Alternatively, the attacker can try to manipulate the encryption-mechanism negotiation between MD and HN and then try to break the negotiated mechanism. How an attacker can manipulate the security-mechanism negotiation has been briefly discussed in
Section 1.3.4. In particular, it depends on how HN and MD protect their negotiation against bidding down.

To recover $E_K_0$ with knowledge of $K_0$ (see Figure 4.7), the attacker must be able to recover $K_0$ and must be able to reconstruct $E_K_0$ from $K_0$ by means of reconstructing the key-establishment process $ke_0$. How the attacker can reconstruct $ke_0$ depends on whether the key-establishment process is static or dynamic (see Definition 1.2.9) and is illustrated in Figure 4.9. In the case of a static key-establishment protocol $ke_0$, the key establishment depends on the secret master key $K_0$ only. In the static case, reconstructing the key-establishment process $ke_0$ is consequently equivalent to knowing the long-term key $K_0$. In the dynamic case, the attacker has to recover the $ke_0$ traffic between MD and HN’s NAP in order to derive $E_K_0$ from $K_0$. Both cases are covered in the following definition:

**Definition 4.2.3** An attacker can **reconstruct** $ke$ if he can recover $E_K$ and $I_K$ from $K$.

If $ke$ is static, an attacker can reconstruct $ke$ if and only if he knows $ke$. If $ke$ is dynamic, the attacker additionally has to recover the $ke$-related part of the control traffic between MD and NAP in order to be able to derive the data-protection keys $E_K$ and $I_K$ from the master session key $K$.

In Figure 4.7, the attacker can recover $K_0$ without knowledge of $E_K_0$ and $I_K_0$ in two ways: either with or without knowledge of $K_1$. The latter case is described in Figure 4.10. To recover $K_0$ in this case, the attacker can try to compromise Eipe_{HN} and read $K_0$ from its memory. The attacker could also compromise AS_{HN} and read $K_0$ from its memory. Alternatively, the attacker can try to recover the plaintext of key-transfer $kt$ of $K_0$ from AS_{HN} to Eipe_{HN} (see Definition 1.2.8). In case the key-derivation function $kd_0$ is the identity, the key $K_0$ itself is transferred in the security context to AS_{DEST}. In this case, the attacker can also try to recover the plaintext of the $S_1$-transfer in order to recover $K_0$. Alternatively, the attacker can try to break $a_0$ (in both directions) and mount a (two sided) man-in-the-middle attack (MiM) against $ka_0$. 

![Figure 4.9: Attack Tree for Subgoal “Reconstruct $ke_0$.”](image-url)
4.2 First-Order—HN as Anchor—SCT with Key Derivation

Figure 4.10: Attack Tree for the Subgoal “Recover $K_0$ without Knowledge of $E K_0$, $I K_0$, and $K_1$.”

**Definition 4.2.4** An attack can break $a_0$ in both directions iff he can impersonate MD to AN and AN to MD throughout the complete message exchanges related to $a_0$.

**Definition 4.2.5** An attacker can mount a two sided man-in-the-middle attack against $k a_0$ iff he can impersonate MD to AN and AN to MD throughout the $k a_0$-related message exchange and can get into possession of master session keys $K_0$ and $K_0^*$, where MD believes that it agreed upon $K_0$ with AN and AN believes it agreed upon $K_0^*$ with MD.

To recover $K_0$ with knowledge of $K_1$ (see Figure 4.7), the attacker must recover $K_1$ and must be able to invert $k d_0$. To recover $K_1$ in this case, the attacker can either be with or without knowledge of $E K_1$ and $I K_1$. The latter case is described in Figure 4.11. In order

Figure 4.11: Attack Tree for the Subgoal “Recover $K_1$ without Knowledge of $E K_1$, $I K_1$, and $K_0$.”

to recover $K_1$ without exploiting any knowledge of other keys, the attacker either must compromise the network components in which $K_1$ is stored or must recover the plaintext of a message in which $K_1$ is transferred from one network component to another, i.e., the transfer of $S_1$ from AS$_{HN}$ to AS$_{DEST}$ or transfer of $K_1$ from AS$_{DEST}$ to EIPE$_{DEST}$. To recover $K_1$ with knowledge of $E K_1$ and $I K_1$ (see Figure 4.7), the attacker must recover $E K_1$ and $I K_1$ and must be able to recover $K_1$ from the two data-protection keys:

**Definition 4.2.6** An attacker can totally break $k e$ if and only if he can recover $K$ from the knowledge of $E K$ and $I K$.
The attacker has two alternatives to recover the encryption key $E^k_1$ without knowledge of $K_1$. He can either intercept traffic encrypted with $e^m_1$ and totally break $e^m_1$, or he can compromise the memory of the encryption end point in DEST and read $E^k_1$ from there. Figure 4.12 describes how an attacker can totally break $e^m_1$. The attacker can either

![Attack Tree for Subgoal “Totally Break $e^m_1$ without $E^k_0$ and without Disabling $e^m_0$.”](image)

try to break whatever $e^m_1$ the networks and MD negotiate in the security-mechanism negotiation phases, or he can try to manipulate the choice of $e^m_1$ and then try to break the resulting mechanism. To manipulate the choice of $e^m_1$, the attacker can manipulate the negotiation between HN and DEST to an element of a specific set of weak encryption mechanisms. Another possibility is to manipulate the selection of DEST and make HN choose a destination network that supports weak encryption mechanisms only. In order to manipulate the selection, the attacker can either jam the NAPs of specific available DESTs leaving only a particular choice to HN, or he can amplify the signal of a particular NAP DEST such that its signal is the strongest received and ends up up front in HN’s list of candidate destination networks.

Similarly the attacker has two alternatives to recover the integrity-protection key $I^K_1$ without knowledge of $K_1$. He can either intercept traffic integrity-protected with $i^m_1$ and totally break $i^m_1$, or he can compromise the memory of the integrity end point in DEST.

\footnote{Note that in this situation the attacker cannot manipulate the negotiation phases between HN and MD, as he does not know the encryption key $E^k_0$.}
and read $IK_1$ from there.

**Definition 4.2.7** An attacker can totally break the integrity-protection mechanism im if he can recover $IK$ from intercepted traffic integrity-protected with im and IK.

Figure 4.13 describes how an attacker can totally break $im_1$ without knowledge of $EK_0$ and without disabling $em_0$.

Figure 4.13: Attack Tree for Subgoal “Totally Break $im_1$ without $EK_0$ and without Disabling $em_0$.”

**4.2.3 Attack Modules**

With the help of the attack trees described in the last section and the ones described in Appendix A, we identified recurring Basic Attack Modules (BAMs), as well as Attack Modules (AMs) that combine the basic attacks to more sophisticated modules. An attacker can combine these attack modules with actions specific to the root attack scenario and the handover initiation type in order to mount attacks targeted to reach his goal in one of the root attack scenarios RAS-1 to RAS-10. While identifying attacks for a root attack scenario is easier using the attack tree notation, identifying recurring attack modules makes it easier to define requirements for the protection against the identified attacks. This is the reason why we change from one form of representation to another. Moreover, as new security
solutions for wireless access networks evolve, small changes to our security model may be
required. The modular description of the attacks allows for replacing single attack modules
with new ones and for deleting particular attack modules without requiring changes to the
threat analysis as a whole.

According to our model of the security mechanisms used in HN and DEST, the master
and data-protection keys are derived in the following way: $K_0$ is computed during the last
key agreement $ka_0$ between MD and AS$_{HN}$. $EK_0$ and $IK_0$ are derived from $K_0$ by the static
or dynamic key-establishment protocol $ke_0$ between MD and EIPE$_{HN}$. $K_1$ is derived from
$K_0$ by means of the key-derivation function $kd_0$. $EK_1$ and $IK_1$ are generated from $K_1$ by
the static or dynamic key-establishment protocol $ke_1$ between MD and EIPE$_{DEST}$. This
translates to basic relations between the keys used before and after handover. An attacker
can try to exploit these key relations to perform the basic attack modules described in the
following. Note that at the end of Appendix A we show how we identified the basic attack
modules in the attack trees.

**BAM-1** An attacker in possession of $EK_0$ and $IK_0$ can recover $K_0$ if he can totally break
the key-establishment process $ke_0$ (see Def. 4.2.6).

**BAM-2** An attacker in possession of $K_0$ can recover $K_1$ if he knows $kd_0$ (and all other
optional input to $kd_0$).

**BAM-3** An attacker in possession of $K_1$ can recover $EK_1$ and $IK_1$ if he knows the static
key-establishment process $ke_0$ or if he can reconstruct the key-establishment traffic of the
dynamic key-establishment process $ke_0$ (see Def. 4.2.3).

**BAM-4** An attacker in possession of $EK_1$ and $IK_1$ can recover $K_1$ if he can totally break
the key-establishment process $ke_1$ (see Def. 4.2.6).

**BAM-5** An attacker in possession of $K_1$ can recover $K_0$ if he can invert the key-derivation
function $kd_0$.

**BAM-6** An attacker in possession of $K_0$ can recover $EK_0$ and $IK_0$ if he knows the static
key-establishment process $ke_0$ or he can reconstruct the key-establishment traffic of a dy-
namic key-establishment process $ke_0$ (see Def. 4.2.3).

An attacker can get into possession of $EK_0$ and $IK_0$ (respectively $EK_1$ and $IK_1$) without
exploiting the key relations by exploiting vulnerabilities of HN (respectively DEST).

**BAM-7** An attacker can recover $EK_0$ and $IK_0$ without exploiting the basic key relations
by any of the following alternatives:

(a) totally breaking $em_0$ and $im_0$ (see Def. 4.2.7 and Def. 4.2.2).

(b) bidding down the $em_0$ and $im_0$ negotiation and then trying to totally break the negotiated
$em_0$ and $im_0$. 
(c) compromising the memory of EIPE$_{HN}$ in which EK$_0$ and IK$_0$ are stored.

(d) breaking $a_0$ and mounting a man-in-the-middle attack against $ka_0$.

BAM-8 An attacker can recover $K_0$ without exploiting the basic key relations by any of the following alternatives:

(a) recovering the plaintext of the $K_0$ transfer from AS$_{HN}$ to EIPE$_{HN}$.

(b) compromising the memory of EIPE$_{HN}$ in which $K_0$ is stored.

(c) compromising the memory of AS$_{HN}$ in which $K_0$ is stored.

BAM-9 An attacker can recover $K_1$ without exploiting the basic key relations by any of the following alternatives:

(a) recovering the plaintext of the $K_1$ transfer from AS$_{HN}$ to EIPE$_{HN}$.

(b) compromising the memory of EIPE$_{HN}$ in which $K_1$ is stored.

(c) compromising the memory of AS$_{HN}$ in which $K_1$ is stored.

(d) recovering the plaintext of the $S_1$ transfer from HN to DEST during the handover procedure.

BAM-10 An attacker can recover EK$_1$ and IK$_1$ without exploiting the basic key relations by any of the following alternatives:

(a) totally breaking $em_1$ and $im_1$ (see Def. 4.2.7 and Def. 4.2.2).

(b) bidding down the $em_1$ and $im_1$ negotiation and then trying to totally break the negotiated $em_1$ and $im_1$.

(c) compromising the memory of EIPE$_{DEST}$ in which EK$_1$ and IK$_1$ are stored.

An attacker can combine the above basic attack modules to build the following attack modules:

AM-1 An attacker can recover EK$_0$ and IK$_0$ by BAM-7. He can then use BAM-1 to recover $K_0$, use BAM-2 to recover $K_1$, and finally recover EK$_1$ and IK$_1$ by exploiting BAM-3.

AM-2 An attacker can recover $K_0$ by BAM-8. He can then use BAM-2 to recover $K_1$ and can therefore recover EK$_1$ and IK$_1$ by exploiting BAM-3.
**AM-3** An attacker can recover $K_1$ by BAM-9. He can then use BAM-3 to recover $EK_1$ and $IK_1$.

**AM-4** An attacker can recover $EK_1$ and $IK_1$ by BAM-10. He can then use the basic key relation BAM-10 to recover $K_1$, use BAM-5 to recover $K_0$, and finally recover $EK_0$ and $IK_0$ by exploiting BAM-6.

**AM-5** An attacker can recover $K_1$ by BAM-9. He can then use BAM-5 to recover $K_0$ and can therefore recover $EK_0$ and $IK_0$ by exploiting BAM-6.

**AM-6** An attacker can recover $K_0$ by BAM-8. He can then use BAM-6 to recover $EK_0$ and $IK_0$.

Figure 4.14 summarizes the BAMs and AMs we consider.

![Figure 4.14: Summary of BAMs and AMs for First-Order Handover with HN as Anchor](image)

The attack modules BAM-1 to BAM-6 exploit the ways the various keys used before and after handover are related. In addition, an attacker can also use a bidding-down attack against the negotiation of the cipher suite $cs_1$ as a module for more sophisticated attacks.

**AM-7** An attacker bids down the negotiation of $em_1$ and $im_1$.

It is important to note that, although the attack trees themselves differ dependent on the handover initiation type, the attack modules we identify in this section can be used as modules for attacks against procedures of any initiation type. In the next section we use the identified attack modules to describe attacks against a network-initiated procedure.
4.2.4 Attacks against a Network-Initiated Procedure

An attacker can combine the above attack modules and other steps specific to the root attack scenario in order to try to achieve the attack goal in the root scenario. In this section, we summarize the attacks against a first-order network-initiated handover procedure with HN as anchor.

Attacks in RAS-1.

A-1 An attacker can use AM-4 to recover $E_{K_0}$ after a handover of MD from HN to DEST has taken place and thus recover the plaintext of intercepted and recorded traffic.

A-2 An attacker can use AM-5 to recover $E_{K_0}$ after a handover of MD from HN to DEST has taken place and thus recover the plaintext of intercepted and recorded traffic.

Attacks in RAS-2.

A-3 An attacker waits until the SCT from HN to DEST takes place and then uses AM-5 to recover $I_{K_0}$ and $E_{K_0}$. He jams NAP$_{DEST}$ such that MD and NAP$_{DEST}$ cannot associate. When MD tries to re-associate with NAP$_{HN}$, he jams the real NAP$_{HN}$ and impersonates NAP$_{HN}$ and EIPE$_{HN}$ with the help of the recovered data-protection keys.

A-4 In a variant of A-3, the attacker disables $em_0$ and $im_0$ before handover and thus does not have to recover the data-protection keys in order to be able to successfully impersonate NAP$_{HN}$ in the above scenario.

In A-3, the attacker gains access to the plaintext of data and control traffic exchanged between MD and HN because a handover to DEST is initiated. As opposed to this, in A-4 the attacker already has access to the plaintext and prevents a handover after which he might not have access to the plaintext of data and control traffic any more.

A-5 An attacker can also try to conduct AM-6 to recover $E_{K_0}$ and $I_{K_0}$ or try to recover $E_{K_0}$ and $I_{K_0}$ directly by BAM-7. As in the last attack, he then has access to the plaintext of data and control traffic exchanged between HN and MD. He can then proceed as above to prevent a handover by simulating a handover failure.

Attacks in RAS-3.

A-6 An attacker can try to use AM-2 to recover $E_{K_1}$.

A-7 An attacker can try to use AM-1 to recover $E_{K_1}$.

A-8 An attacker can use AM-7 and then try to totally break $em_1$ to recover $E_{K_1}$. 
Attacks in RAS-4.

A-9 An attacker recovers $E_0$ and $I_0$ with the help of BAM-7. He then intercepts the control traffic exchanged between MD and HN. On intercepting a handover command to $N_{dest}$, he uses AM-1 to recover $E_1$ and $I_1$. He then jams the real $N_{dest}$ and impersonates $N_{dest}$ and $E_{dest}$ to MD with the help of the recovered data-protection keys when MD tries to associate.

A-10 An attacker recovers $K_0$ using BAM-8 before handover. He then intercepts the control traffic exchanged between MD and HN. On intercepting a handover command to $N_{dest}$, he uses AM-2 to recover $E_1$ and $I_1$ and proceeds as in A-9.

A-11 An attacker bids down the $em_1, im_1$ negotiation (AM-7) until $em_1$ and $im_1$ are disabled. He then jams the real $N_{dest}$ and impersonates $N_{dest}$ to MD.

A-12 An attacker waits until HN transfers the security context to DEST and then recovers $I_k$ and $E_k$ by AM-2. He then proceeds as in A-9.

Note that in A-12, the attacker has to get knowledge of DEST without having access to the key. This is possible if HN does not encrypt control traffic (in this case, the attacker can intercept the plaintext of the handover command). If HN encrypts control traffic, the attacker can try to guess DEST from MDs location.

A-13 An attacker breaks the initial authentication protocol $a_0$ and mounts a man-in-the-middle attack against MD and HN and thus gets hold of master keys $K_0$ and $K_{0*}$, where MD believes that it agreed upon $K_0$ with HN and HN believes it agreed upon $K_{0*}$ with MD. Then the attacker can successfully impersonate MD to HN and HN to MD. If now MD and the attacker move, and HN can no longer serve the attacker, the attacker can use AM-2 to generate the data-protection keys that enable him to impersonate DEST to MD and MD to DEST.

Attacks in RAS-5.

A-14 An attacker recovers $E_0$ and $I_0$ with the help of BAM-7 and uses them to fake a handover command to MD. Additionally, he uses AM-1 to recover $E_1$ and $I_1$ according to the DEST identity he included in the fake handover command. He then impersonates $N_{dest}$ and $E_{dest}$ to MD with the help of the recovered data-protection keys when MD tries to associate to $N_{dest}$.

A-15 An attacker recovers $E_0$ and $I_0$ with the help of AM-6. He then uses $E_0$ and $I_0$ to fake a handover command and uses AM-2 to generate $E_1$ and $I_1$ according to the DEST identity he included in the fake handover command. He then proceeds as in A-14.

A-16 An attacker disables $em_0$ and $im_0$ and sends a fake handover command to MD. He uses AM-2 to recover $K_0$ and generate $E_1$ and $I_1$. He then proceeds as in A-14.
4.2 First-Order—HN as Anchor—SCT with Key Derivation

Attacks in RAS-6.

A-17 An attacker can try to recover $K_0$ using BAM-8 and then use AM-2 to recover $IK_1$.

A-18 An attacker can try to use AM-1 to recover $EK_0$ and $IK_0$ and therefore derive $IK_1$.

A-19 An attacker can try to use AM-7 and then try to totally break $im_1$ to recover $IK_1$.

Attacks in RAS-7.

A-20 This attack is restricted to hard handover procedures in which HN waits for MD to fall back for a certain time after MD has disassociated from HN. An attacker has managed to recover $EK_0$ and $IK_0$ by BAM-7 or by AM-6 or by AM-5 and to suppress the handover complete message. He then impersonates MD to HN within the fall-back time.

A-21 In case the attacker managed to disable $em_0$ and $im_0$ during the initial negotiation, it is sufficient that the attacker suppresses the handover complete message in order to impersonate MD to HN in the scenario of A-20.

Attacks in RAS-8.

A-22 An attacker can try to disable $em_1$ and $im_1$ by bidding them down to no encryption and no integrity protection (AM-7). He can then jam the real MD and impersonate MD to $NAP_{dest}$ and $EIPE_{dest}$.

A-23 An attacker can try to use AM-1 or AM-2 to recover $IK_1$ and $EK_1$. He can also get into the possession of $IK_1$ and $EK_1$ by recovering $K_1$ before MD and $NAP_{dest}$ associate or even read $EK_1$ and $IK_1$ from $EIPE_{dest}$’s memory. With the help of the data-protection keys and by jamming the real MD, he can then impersonate MD to $NAP_{dest}$ an $EIPE_{dest}$.

A-24 If an attacker has managed to disable $em_0$ and $im_0$ and to impersonate MD to HN, then he can try to bid down the $em_1, im_1$ negotiation to disable $em_1$ and $im_1$.

If A-24 is successful, an initial service theft against HN leads to a service theft against DEST.

A-25 If an attacker can recover $EK_0$ and $IK_0$ or $K_0$ and manages to impersonate MD to HN before handover, then he can use AM-1 respectively AM-2 to recover $EK_1$ and $IK_1$ and can thus continue to impersonate MD to DEST.
Attacks in RAS-9.

A-26 An attacker can try to jam the association between MD and DEST. He can try to make HN detect a handover reason, although there is none, and then let the ongoing connection drop. He can replay an old handover command to MD. Alternatively, he can jam the security-suite negotiation between MD and HN or MD and DEST, or he can try to bid down the negotiation such that it fails.

Attacks in RAS-10.

A-27 This attack only applies to hard handover: An attacker can try to make HN reserve resources for MD longer than necessary by suppressing the handover-complete message sent to HN by DEST.

A-28 An attacker can try to make HN discover a handover reason although there is none. In order to do this, the attacker can impersonate NAP\textsubscript{DEST} or amplify the signal of an existing network access point.

Attacks in RAS-11.

A-29 An attacker can try to send fake handover requests to DEST in order to use up sources in DEST. He can furthermore engage DEST in the security-mechanism negotiation.

Table 4.1 gives an overview on which of the attacks A-1 to A-25 make use of which attack modules.\footnote{The Denial of Service (DoS) attacks A-26 to A-29 are considered part of the general DoS protection of a wireless network, which is out of the scope of this work.}

To avoid repetition, we refrain from giving a detailed description of the potential attacks against mobile-initiated first-order handover procedures with HN as anchor at this point. We will instead point out the differences on mobile-initiated procedures after describing the general case of a \( k \)-th-order \((k \geq 1)\) handover with HN or FN as anchor in Section 4.3.4.

4.2.5 Requirements and Recommendations for SCT with Key Derivation

In this section, we define requirements to secure a first-order handover procedure with HN as anchor. Requirements R-1 to R-4 address security problems that arise from the use of weak security mechanisms between MD and HN before a first-order handover. Requirements R-5 and R-6 address problems arising from the use of weak mechanisms after a first-order handover. The requirements R-7 and R-8 protect the handover procedure messages themselves and prevent attacks arising from faked handover requests and handover commands.

Additionally, we specify recommendations on what further measures wireless networks should use to protect themselves. These recommendations cannot be addressed by the handover procedure itself (see RC-1, RC-2).
Table 4.1: Overview on Attacks and Attack Modules

<table>
<thead>
<tr>
<th>Attack Module</th>
<th>Used in Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM-3</td>
<td>A-12</td>
</tr>
<tr>
<td>AM-4</td>
<td>A-1</td>
</tr>
<tr>
<td>AM-5</td>
<td>A-2, A-3, A-20</td>
</tr>
<tr>
<td>AM-6</td>
<td>A-5, A-15, A-20</td>
</tr>
<tr>
<td>BAM-7</td>
<td>A-5, A-9, A-14, A-20</td>
</tr>
<tr>
<td>BAM-8</td>
<td>A-10, A-17</td>
</tr>
<tr>
<td>BAM-10</td>
<td>A-8</td>
</tr>
<tr>
<td>Attacker disables $em_0$, $im_0$</td>
<td>A-4, A-21, A-24</td>
</tr>
<tr>
<td>Attacker mounts MiM against $ka_0$</td>
<td>A-13</td>
</tr>
<tr>
<td>Attacker breaks $a_0$</td>
<td>A-13</td>
</tr>
<tr>
<td>Attacker fakes handover request</td>
<td>A-29</td>
</tr>
</tbody>
</table>

We will first define and motivate the new security requirements and then show how these requirements address the attacks identified in the last section. In particular we show that a handover procedure with SCT that meets our requirements is protected against the attack modules listed on the right side of Table 4.1 and thus against the attacks A-1 to A-25.

Due to a first order handover, DEST may suffer from the use of a disabled $em_0$ or $im_0$, a broken initial authentication or key agreement, a key-establishment process $ke_0$ that is vulnerable to BAM-1, or a key-derivation function $kd_0$ that is vulnerable to BAM-2 or BAM-5. It is therefore crucial for DEST to base its handover decision on the initial security suite $ss_0$, as well as the key-derivation function $kd_0$, by which $K_1$ was derived and we require:

R-1 DEST shall base its decision on whether to accept or refuse a given handover request on (1) the cipher suite $cs_0 = (ke_0, em_0, im_0)$ previously used between MD and HN; (2) the initial authentication protocol $a_0$; (3) the initial key-agreement protocol $ka_0$; and (4) the key-derivation function $kd_0$.

This allows DEST to protect itself by refusing handover if it deems one of the mechanisms used before handover to be too weak. For example, DEST can refuse handover if “no encryption” and “no integrity protection” was used between MD and HN before handover and protect itself against A-4, A-21, and A-24.

By means of R-1, DEST can detect the use of a key-derivation function it does not deem to provide an adequate security level. As opposed to this, requirements R-2 and R-3 specify what properties a key-derivation function should ideally have.
R-2 Knowledge of a master session key $K_1$, used between MD and DEST after handover, shall not reveal any information on any previously used master session key $K_0$ to anyone except MD and HN.

In order to protect against attacks using the basic attack module BAM-2, we require:

R-3 Knowledge of a previously used master session key $K_0$ shall not reveal any information on $K_1$ to anyone except MD and DEST.

Requirements R-2 and R-3 specify properties of a good choice of a key-derivation function, while R-1 allows DEST to protect itself even if HN and MD choose to use a key-derivation function DEST does not trust.

R-4 The security context shall include information on the lifetime of the initial master session key $K_0$.

This, for example, allows DEST to have considerations on the key lifetime be part of its handover decision. It furthermore allows HN to pass on its restrictions on key lifetimes to DEST.

It is crucial for HN to influence the selection of mechanisms to be used after handover as otherwise it may suffer from impersonation attacks due to the use of a weak cipher suite $cs_1$. The use of a weak or (partially or totally) breakable cipher suite after handover, for example, is necessary in order to mount the impersonation attacks A-11 and A-22 against DEST and MD as well as the attack A-1 against the confidentiality of the air interface before handover. Moreover, a reconstructible $ke_1$ may lead to A-12.

R-5 The negotiation of the cipher suite $cs_1 = (em_1, im_1, ke_1)$ shall enforce compliance with policies set by MD, DEST, and HN.

We identified “bidding down” of the negotiation of the cipher suite $cs_1$ as a basic attack module that can be used to mount the attacks A-8, A-11, A-19, A-22, and A-24. In order to protect against these attacks, we require:

R-6 The negotiation of the cipher suite $cs_1 = (em_1, im_1, ke_1)$ shall be protected against bidding-down attacks.

In order to protect DEST from fake handover requests as used in A-29 and from recovery of the transferred key $K_1$ (BAM-9 used in A-2 to A-4, A-12, and A-20), we require:

R-7 The security-context transfer of related keys between HN and DEST shall be encrypted and integrity-protected (including replay protection) and shall provide a proof of its origin (HN) to DEST.

In order to protect MD from fake handover commands as used in A-14, A-15, A-16, and A-26 we require:
4.2 First-Order—HN as Anchor—SCT with Key Derivation

**R-8** The handover command sent from HN to MD shall be integrity-protected (including replay protection) and provide a proof of its origin (HN) to MD.

DEST may suffer from attacks that make use of the basic attack modules BAM-8 and (c) of BAM-7. These basic attack modules make use of vulnerabilities of the network components and unprotected communication between network components in HN. The handover procedure itself cannot protect against these basic attack modules. A key-derivation method that meets R-3 prevents attacks that use BAM-8 as module in AM-2. The only identified attack that uses BAM-8 or BAM-7 directly is A-5, from which HN, but not DEST, suffers. With an eye on the potentially evolving attacks, we recommend:

**RC-1** A destination network shall carefully consider entering a handover agreement with HN that does not protect the memory of its EIPE$_{HN}$ and/or AS$_{HN}$ against unauthorized access (including physical access), that does not protect the confidentiality and integrity of key transfers from AS$_{HN}$ to EIPE$_{HN}$, or that keeps $K_0$ in EIPE$_{HN}$’s memory longer than necessary.

In particular, DEST shall not enter into handover agreements with HN if R-3 is not met and RC-1 cannot be ensured.

**RC-2** HN shall carefully consider entering handover agreements with destination networks that do not protect the memory of their EIPE$_{DEST}$ and/or AS$_{DEST}$ against unauthorized access (including physical access) that do not protect the confidentiality and integrity of key transfers from AS$_{DEST}$ to EIPE$_{DEST}$, or that keep $K_1$ in EIPE$_{DEST}$’s memory longer than necessary.

In particular, HN shall not enter agreements with DESTs that do not meet RC-2 if it could be held responsible for the impersonation attack A-12. Note that the other attacks with impact on HN that make use of the alternatives (a), (b), or (c) in BAM-9 or (c) in BAM-10 can be made impossible by the use of a key-derivation function $kd_0$ that meets R-2.

In order to prevent an attacker from exploiting HN’s channel reservation for a denial of service attack by suppressing the handover-complete message (see A-27), HN has to carefully set fall-back times on hard handover. However, this and other measures to protect the handover participants against DoS attacks are considered out of scope of this work.

As illustrated in Table 4.1 above, all of the attacks A-1 to A-25 make use of at least one of the following:

(a) one of the attack modules AM-1 to AM-6;
(b) one of the basic attack modules BAM-7, BAM-8, or BAM-10 directly;
(c) disable $em_0$ and $im_0$;
(d) mount a man-in-the-middle attack against the key-agreement protocol $ka_0$;
(e) break the initial authentication protocol $a_0$;
Attack Module | Adressed by
--- | ---
BAM-1 (AM-1) | R-1 DEST bases its decision on $ke_0$
BAM-2 (AM-1, AM-2) | R-3
BAM-3 | not addressed\(^4\)
BAM-4 (AM-4) | R-5 Policies of HCN during $ke_1$ negotiation enforced
BAM-5 (AM-4, AM-5) | R-2
BAM-6 | not addressed\(^4\)
BAM-7 (AM-1) | R-1 DEST bases its decision on $em_0$, $im_0$
BAM-8 (AM-2, AM-6) | RC-1
BAM-9 (AM-3, AM-5) | RC-2 and R-7
BAM-10 (AM-4) | R-5 Policies of HCN during $em_1$ and $im_1$ negotiation enforced
AM-7 | R-6
Attacker disables $em_0$, $im_0$ | R-1
Attacker breaks $a_0$ | R-1
Attacker mounts MiM against $ka_0$ | R-1
Attacker fakes handover command | R-8 and R-1
Attacker fakes handover request | R-7 and R-1
Attacker fakes handover indication | R-8 and R-1

Table 4.2: Attack Modules and Requirements

(f) fake a handover request;

(g) fake a handover command.

In order to protect against A-1 to A-25, and any other attacks built from these modules, it is therefore sufficient to protect against (a) – (g).

The attack modules AM-1 to AM-6 each make use of at least two BAMs. In order to protect against an AM, it is therefore sufficient to protect against one of the two BAMs it uses.

Table 4.2 illustrates, how our requirements protect against the basic attack modules, as well as the other modules listed above.

A first-order handover procedure that meets the above requirements is secure against the attacks A-1 to A-26. In the next chapter, we present our history-enriched, policy-based approach that meets R-1, R-2, and R-4 to R-8 in whole and R-3 in part.

\(^4\)BAM-3 and BAM-6 are not used in any of the attacks on their own.
4.3 $k$-th-order Handover with HN or FN as Anchor and SCT with Key Derivation

In this section, we generalize the root attack scenarios, attack modules, attacks, and requirements from first-order handover with HN as anchor to the general case of a $k$-th-order ($k \geq 1$) handover with HN or FN as anchor. We proceed as in the first-order case and start by describing the root attack scenarios of an attacker. As in the first-order handover case, the attack root attack scenarios are independent of the initiation type of the handover. However, as we will see later on, some of the attacks an attacker can mount in order to achieve his goal in one of the root attack scenarios depend on the type of the handover initiation.

4.3.1 Root Attack Scenarios

The root attack scenarios described for the first-order handover with HN as anchor can easily be generalized to the case of a $k$-th-order handover from a source network SRC$_k$ to a destination network DEST$_k$ with HN or FN as anchor. Differences mainly arise from the fact that an attacker can try to exploit a $k$-th-order handover procedure in order to mount attacks against any of the previously serving networks SRC$_j$ (1 $\leq j \leq k$). It is also important to note that the root attack scenarios as well as the attacks themselves can be described independently from the anchor type of the handover procedure.

**RAS*-1** An attacker recovers the plaintext of encrypted data or control traffic he intercepted and recorded on the air interface between MD and SRC$_j$ for some 1 $\leq j \leq k$ after a $k$-th-order handover of MD from SRC$_k$ to DEST$_k$ has taken place (violates confidentiality goals of MD and SRC$_j$ 1 $\leq j \leq k$).

**RAS*-2** An attacker recovers the plaintext of data or control traffic sent by MD by impersonating SRC$_k$ on a simulated handover failure (violates confidentiality goals of MD and SRC$_k$).

**RAS*-3** An attacker recovers the plaintext of data or control traffic exchanged between MD and an authorized DEST$_k$ after a handover of MD from HN to DEST$_k$ taken place (violates confidentiality goals of MD and DEST$_k$).

**RAS*-4** An attacker gains access to the plaintext of data or control traffic sent by MD by impersonating DEST$_k$ to MD on an actual handover of MD from SRC$_k$ to DEST$_k$.

**RAS*-5** An attacker gains access to the plaintext of data or control traffic sent by MD by simulating a handover from SRC$_k$ to DEST$_k$ and impersonating DEST$_k$ to MD.

**RAS*-6** An attacker manipulates data or control traffic between MD and an authorized DEST$_k$ after handover (violates integrity goal of MD and DEST$_k$).
**RAS*-7** An attacker tries to gain access to $\text{SRC}_k$’s network on behalf of a victim MD exploiting an actual handover procedure of MD from $\text{SRC}_k$ to $\text{DEST}_k$.

**RAS*-8** An attacker tries to gain access to $\text{DEST}_k$’s network on behalf of a victim MD exploiting an actual handover procedure of the victim MD from $\text{SRC}_k$ to $\text{DEST}_k$.

**RAS*-9** An attacker prevents a legitimate MD to continuously use $\text{SRC}_k$’s or $\text{DEST}_k$’s service by interfering with the handover procedure (DoS against MD).

**RAS*-10** An attacker uses the handover procedure to block resources in $\text{SRC}_k$ (DoS against $\text{SRC}_k$).

**RAS*-11** An attacker exploits the handover procedure to overload $\text{DEST}_k$ (DoS against $\text{DEST}_k$).

### 4.3.2 Attack Modules

In this section, we generalize the attack modules described for first-order handover procedures with HN as anchor to attack modules against a $k$-th-order handover ($k \geq 1$) with HN or FN as anchor. As before we denote the anchor network (HN or FN) with AN. Although the attacks against a handover procedure differ with the initiation type, we identified only attack parts as modules that are independent of the initiation type of the handover procedure. In the $k$-th-order handover, the use of any weak security mechanisms between MD and any of the previously serving networks $\text{AN}(= \text{SRC}_1), \text{SRC}_2, \ldots, \text{SRC}_k$, or the Handover Controlling Network (HCN) are potential threats to the participants of a handover procedure. As described in Section 3.2.3.2, we assume that HCN generates the master key $K_k$ transferred in the security context on a $k$-th-order handover from $K_0$ (AN-controlled case, HN-controlled case) or $K_{k-1}$ (SRC-controlled case):

\[
K_i = kd_{i-1}(K_{i-1}, \text{[optional other parameters]})
\]  

(4.1)

in the SRC-controlled case and

\[
K_i = kd_0(K_0, \text{[optional other parameters]})
\]  

(4.2)

in the HN-controlled and AN-controlled cases.

The initial master session key $K_0$ agreed upon between MD and AN has been generated during the initial authentication $(r)a_0$ by means of the (roaming) key-agreement protocol $(r)ka_0$.

The way HCN derives $K_i$ from $K_{i-1}$ and the way the data-protection keys are derived from the master keys lead to the following basic attack modules:

**BAM*-1** For $0 \leq i \leq k$, an attacker in possession of $E_{K_i}$ and $I_{K_i}$ can recover $K_i$ if he can totally break the key-establishment process $ke_i$ (see Def. 4.2.6).
4.3 \( k \)-th-order Handover—SCT with Key Derivation

**BAM*-2** An attacker in possession of \( K_j, j \leq k \) can recover \( K_k \) if

(i) he knows \( kd_{j+1}, \ldots, kd_{k-1} \) and can therefore subsequently compute \( K_j, \ldots, K_k \) (SRC-controlled case).

(ii) he knows and can invert \( kd_0 \) to obtain \( K_0 \) from \( K_j \) and can therefore compute \( K_k \) from \( K_0 \) (HN-controlled case, AN-controlled case).

**BAM*-3** An attacker in possession of \( K_k \) can recover \( K_j, 0 \leq j \leq k - 1 \) if

(i) he knows and can invert \( kd_{j+1}, \ldots, kd_{k-1} \) and can therefore compute \( K_{k-1}, \ldots, K_j \) (SRC-controlled case).

(ii) he knows and can invert \( kd_0 \) to obtain \( K_0 \) from \( K_k \) and can therefore compute \( K_j \) from \( K_0 \) (HN-controlled case, AN-controlled case).

**BAM*-4** For \( 0 \leq i \leq k \), an attacker in possession of \( K_i \) can recover \( EK_i \) and \( IK_i \) if he knows the static key-establishment process \( ke_i \) or if he can reconstruct the key-establishment traffic of the dynamic key-establishment process \( ke_i \) (see Def. 4.2.3).

An attacker can get into possession of \( EK_i \) and \( IK_i \) without exploiting the key relations by exploiting vulnerabilities of mechanisms used between MD and DEST\( ^i \) or a missing protection of keys by DEST\( ^i \).

**BAM*-5** An attacker can recover \( EK_i \) and \( IK_i \) (\( 0 \leq i \leq k \)) without exploiting the key relations by any of the following alternatives:

(a) totally breaking \( em_i \) and \( im_i \) (see Def. 4.2.7 and Def. 4.2.2).

(b) bidding down the \( em_i \) and \( im_i \) negotiation and then trying to totally break the negotiated \( em_i \) and \( im_i \).

(c) compromising the memory of EIPE\( ^{DEST_i} \) in which \( EK_i \) and \( IK_i \) are stored.

**BAM*-6** An attacker can recover \( K_i \) (\( 1 \leq i \leq k \)) without exploiting the basic key relations by any of the following alternatives:

(a) recovering the plaintext of the \( K_i \) transfer from AS\( ^{DEST_i} \) to EIPE\( ^{DEST_i} \).

(b) compromising the memory of EIPE\( ^{DEST_i} \) in which \( K_i \) is stored.

(c) compromising the memory of AS\( ^{DEST_i} \) in which \( K_i \) is stored.

(d) compromising the memory of AS\( ^{HCN} \) in which \( K_i \) is stored.

(e) recovering the plaintext of the \( S_i \) transfer from HCN to DEST\( ^i \) during the handover procedure.
BAM*-7 An attacker can recover $K_0$ without exploiting the basic key relations by any of the following alternatives:

(a) recovering the plaintext of the $K_0$ transfer from $A_{SN}$ to $EIP_{AN}$.

(b) (if $HN$ generates $K_0$ during $(r)ka_0$) compromising the memory of $A_{SN}$.

(c) (if $HN$ generates $K_0$ during $(r)ka_0$) recovering the plaintext of the $K_0$ transfer from $HN$ to $AN$.

(d) compromising the memory of $EIP_{AN}$ in which $K_0$ is stored.

(e) breaks $a_0$ and mounts MiM against $ka_0$.

An attacker can combine the above basic attack modules to build the following attack modules:

AM*-1 An attacker can recover $EK_j$ and $IK_j$ for some $0 \leq j \leq k - 1$ by BAM*-5. He can then use BAM*-1 to recover $K_j$, use BAM*-2 to recover $K_k$, and finally recover $EK_k$ and $IK_k$ by means of BAM*-4.

AM*-2 An attacker can recover $K_j$ for some $1 \leq j \leq k - 1$ by BAM*-6 or recover $K_0$ by BAM*-7. He can then use BAM*-2 to recover $K_k$ and can therefore recover $EK_k$ and $IK_k$ by means of BAM*-4.

AM*-3 An attacker can recover $K_k$ by BAM*-6. He can then use BAM*-4 to recover $EK_k$ and $IK_k$.

AM*-4 An attacker can recover $EK_k$ and $IK_k$ by BAM*-5. He can then use the basic attack module BAM*-1 to recover $K_k$, use BAM*-3 to recover $K_j$, and finally recover $EK_j$ and $IK_j$ by means of BAM*-4.

AM*-5 An attacker can recover $K_k$ by BAM*-6. He can then use BAM*-3 to recover $K_j$ and can therefore recover $EK_j$ and $IK_j$ by means of BAM*-4.

AM*-6 An attacker can recover $K_j$ by BAM*-6. He can then use BAM*-4 to recover $EK_j$ and $IK_j$.

Figure 4.15 gives an overview on the attack modules described so far. New attacks due to handover procedures cannot only arise from the security-context transfer of related keys, but also from the way the cipher suite to use after handover is negotiated. We identified the following additional attack module:

AM*-7 An attacker can bid down the $em_k$ and $im_k$ negotiation if he can recover $EK_{k-1}$ and $IK_{k-1}$ by BAM*-5 or if he can disable $em_{k-1}$ and $im_{k-1}$.
4.3 \textit{k}-th-order Handover—SCT with Key Derivation

4.3.3 Attacks against a Network-Initiated Procedure

An attacker can combine the above attack modules and other steps specific to the root attack scenario to try to achieve his goals in the scenario.

Attacks in RAS*-1.

\textbf{A*-1} An attacker can use AM*-4 to recover EK$_j$ from EK$_k$ and IK$_k$ after a handover of MD from SRC$_k$ to DEST$_k$ has taken place and thus recover the plaintext of intercepted and recorded traffic.

\textbf{A*-2} An attacker can use AM*-5 to recover EK$_j$ from K$_k$ after a handover of MD from SRC$_k$ to DEST$_k$ has taken place and thus recover the plaintext of intercepted and recorded traffic.

Attacks in RAS*-2.

\textbf{A*-3} An attacker waits until the security-context transfer (including K$_k$) from HCN to DEST$_k$ takes place and then uses AM*-5 to recover IK$_{k-1}$ and EK$_{k-1}$. He jams NAP$_{DEST_k}$ such that MD and NAP$_{DEST_k}$ cannot associate. When MD tries to re-associate with NAP$_{SRC_k}$, he jams the real NAP$_{SRC_k}$ and impersonates NAP$_{SRC_k}$ and EIPE$_{SRC_k}$ with the help of the recovered data-protection keys to MD.

\textbf{A*-4} In a variant of the above attack, the attacker disables em$_{k-1}$ and im$_{k-1}$ before handover and thus does not have to recover the data-protection keys in order to be able to successfully impersonate NAP$_{SRC_k}$ and EIPE$_{SRC_k}$ in the above scenario.
In A*-3, the attacker gains access to the plaintext of data and control traffic exchanged between MD and SRC\(_k\) because a handover to DEST\(_k\) is initiated. As opposed to this, in A*-4 the attacker already has access to the plaintext and prevents a handover after which he might not have access to the plaintext of data and control traffic any more.

**A*-5** An attacker can also try to conduct AM*-6 to recover EK\(_{k-1}\) and IK\(_{k-1}\) or try to recover EK\(_{k-1}\) and IK\(_{k-1}\) directly by BAM*-5. He can also try to recover EK\(_{k-1}\) and IK\(_{k-1}\) by AM*-2 or AM*-1. As in the last attack, he then has access to the plaintext of data and control traffic exchanged between SRC\(_k\) and MD. He can then proceed as in A*-3 in order to prevent a handover by simulating a handover failure.

**Attacks in RAS*-3.**

**A*-6** An attacker can use AM*-2 to recover EK\(_k\).

**A*-7** An attacker can use AM*-1 to recover EK\(_k\).

**A*-8** An attacker can use AM*-7 and then try to totally break em\(_k\) to recover EK\(_k\).

**Attacks in RAS*-4.**

**A*-9** An attacker recovers EK\(_{k-1}\) and IK\(_{k-1}\) with the help of BAM*-5. He then intercepts the control traffic exchanged between MD and SRC\(_k\). On intercepting a handover command to NAP\(_{DEST_k}\), he uses AM*-1 to recover EK\(_k\) and IK\(_k\). He then jams the real NAP\(_{DEST_k}\) and impersonates NAP\(_{DEST_k}\) and EIPE\(_{DEST_k}\) when MD tries to associate.

**A*-10** An attacker recovers K\(_j\) using BAM*-6 before handover. He then intercepts the control traffic exchanged between MD and SRC\(_k\). On intercepting a handover command to NAP\(_{DEST_k}\), he uses AM*-2 to recover EK\(_k\) and IK\(_k\) and proceeds in the same way as in A*-9.

**A*-11** An attacker bids down the em\(_k\), im\(_k\) negotiation (AM*-7) until em\(_k\) and im\(_k\) are disabled. He then intercepts the control traffic exchanged between MD and SRC\(_k\). On intercepting a handover command to NAP\(_{DEST_k}\), he jams the real NAP\(_{DEST_k}\) and impersonates NAP\(_{DEST_k}\) to MD.

**A*-12** An attacker waits until HCN transfers the security context to DEST\(_k\) and then recovers IK\(_k\) and EK\(_k\) by AM*-3. He then proceeds as in A*-9.

Note that in A*-12, the attacker has to get knowledge of DEST\(_k\) without having access to the data-protection keys. This is possible if SRC\(_k\) does not encrypt control traffic (in this case, the attacker can intercept the plaintext of the handover command). If SRC\(_k\) encrypts control traffic, the attacker can try to guess DEST\(_k\) from MD’s location.
4.3 \( k \)-th-order Handover—SCT with Key Derivation

A*-13 Assume an attacker breaks \( a_0 \) and mounts a a man-in-the-middle attack against \( k_a_0 \) to get into possession of \( K_0 \) and \( K'_0 \) and thus gets hold of master keys \( K_{k-1} \) and \( K'_{k-1} \), where MD believes that it agreed upon \( K_{k-1} \) with \( SRC_k \) and \( SRC_k \) believes it agreed upon \( K'_{k-1} \) with MD. Then the attacker can successfully impersonate MD to \( SRC_k \) and \( SRC_k \) to MD. If now MD and the attacker move, and \( SRC_k \) can no longer serve the attacker, the attacker tries to use AM*-2 to generate the data-protection keys that enable him to impersonate \( DEST_k \) to MD and MD to \( DEST_k \).

Attacks in RAS*-5.

A*-14 An attacker recovers \( EK_{k-1} \) and \( IK_{k-1} \) with the help of BAM*-5 and uses them to fake a handover command to MD. Additionally, he uses AM*-1 to recover \( EK_k \) and \( IK_k \) according to the \( DESt_k \) identity he included in the fake handover command. He then impersonates \( NAP_{DEStk} \) and \( EIPE_{SRCk} \) to MD when it tries to associate with it.

A*-15 An attacker recovers \( EK_{k-1} \) and \( IK_{k-1} \) with the help of AM*-6 or AM*-2 or AM*-1. He then uses \( EK_{k-1} \) and \( IK_{k-1} \) to fake a handover command and uses AM*-2 to generate \( EK_k \) and \( IK_k \) according to the \( NAP_{DEStk} \) identity he included in the fake handover command. He then proceeds as in A*-14.

A*-16 An attacker disables \( em_{k-1} \) and \( im_{k-1} \) and sends a fake handover command to MD. He uses AM*-2 to recover \( K_j \) and generate \( EK_k \) and \( IK_k \). He then proceeds as above.

Attacks in RAS*-6.

A*-17 An attacker can recover \( K_j \) using BAM*-6 and then use AM*-2 to recover \( IK_k \).

A*-18 An attacker can use AM*-1 to recover \( EK_j \) and \( IK_j \) and therefore derive \( IK_k \).

A*-19 An attacker can use AM*-7 and then try to totally break \( im_k \) to recover \( IK_k \).

Attacks in RAS*-7.

A*-20 This attack is restricted to hard handover procedures in which \( SRC_k \) waits for MD to fall back for a certain time after MD has disassociated from \( SRC_k \). An attack recovers \( EK_{k-1} \) and \( IK_{k-1} \) by BAM*-5 or by AM*-6, or AM*-5, or AM*-2, or AM*-1. He suppresses the handover-complete message and impersonates MD to \( SRC_k \) during the fall-back time.

A*-21 In case the attacker manages to disable \( em_{k-1} \) and \( im_{k-1} \) during the negotiation, it is sufficient that the attacker suppresses the handover-complete message in order to impersonate MD to \( SRC_k \).
Attacks in RAS*-8.

A*-22 An attacker can try to disable $em_k$ and $im_k$ by bidding them down to no encryption and no integrity protection (AM*-7). He can then jam the real MD and impersonate MD to DEST$_k$ with it.

A*-23 An attacker can try to use AM*-1 or AM*-2 to recover $IK_k$ and $EK_k$. He can also get into the possession of $IK_k$ and $EK_k$ by recovering $K_k$ before MD and NAP$_{DEST}$ associate or even read $EK_k$ and $IK_k$ from EIPE$_{DEST}$'s memory. With the help of the data-protection keys and with the help of jamming the real MD, he can then impersonate MD to DEST$_k$.

A*-24 If an attacker manages to disable $em_{k-1}$ and $im_{k-1}$ and manages to impersonate MD to SRC$_k$, he can then try to bid down the $em_k, im_k$ negotiation to disable $em_k$ and $im_k$.

If A*-24 is successful, an initial service theft against SRC$_k$ leads to a service theft against DEST$_k$.

A*-25 If an attacker can recover $EK_{k-1}$ and $IK_{k-1}$ or $K_{k-1}$ and manages to impersonate MD to SRC$_k$ before handover, then he can use AM*-1 or AM*-2 to recover $EK_k$ and $IK_k$ and can thus continue to impersonate MD to DEST$_k$.

Attacks in RAS*-9.

A*-26 An attacker can jam the association between MD and DEST$_k$, he can make SRC$_k$ detect a handover reason although there is none and then let the ongoing connection drop. He can replay an old handover command to MD, jam the security suite negotiation between MD and SRC$_k$ or MD and DEST$^k$, or bid down the negotiation to fail.

Attacks in RAS*-11.

A*-27 An attacker sends fake handover requests to DEST$_k$ in order to use up resources in DEST$_k$. He can furthermore engage DEST$_k$ in the security-mechanism negotiation.

Attacks in RAS*-10. The next attack only applies to hard handover procedures.

A*-28 An attacker makes SRC$_k$ reserve resources for MD longer than necessary by suppressing the handover-complete message sent from DEST$_k$ to SRC$_k$.

A*-29 An attacker can make SRC$_k$ discover a handover reason although there is none. In order to achieve this, the attacker can impersonate the network access point of a destination network with which HCN has a handover agreement, or amplify the signal of an existing network access point.
4.3.4 Attacks against a Mobile-Initiated Procedure

In order to avoid unnecessary repetitions, we analyze SCT for both types of mobile-initiated handover procedures at once. We detail the case that HCN is notified by MD over SRC_k and note the changes for the case that HCN is notified by MD over DEST_k within parentheses. Moreover, we describe the first-order and the subsequent handover case at once and do not explicitly distinguish between HN or FN as anchor, that is, we describe the k-th-order (k ≥ 1) handover case with an arbitrary AN.

As already mentioned, we described the root attack scenarios and attack modules independent of the handover initiation type. Moreover, some of the attacks described in the last section, namely A*-1 to A*-8, A*-13, A*-17 to A*-25, and A*-28 are not specific to the type of the handover initiation and are thus to be considered for the mobile-initiated case as well. However, the attacks A*-9 to A*-12, A*-14 to A*-16, A*-26, A*-27, and A*-29 are specific to network-initiated handover procedures, as they exploit messages specific to this procedure type.

The attacks A*-9 to A*-12, A*-26, A*-27, and A*-29 can easily be adapted to the mobile-initiated case in the following way:

**A*-mob-9** An attacker recovers EK_j and IK_j with the help of BAM*-5. He then intercepts the control traffic exchanged between MD and SRC_k. On intercepting a handover-indication message sent from MD to HCN over SRC_k (over DEST_k), indicating handover to NAP_{DEST_k}, he uses AM*-1 to recover EK_k and IK_k. He then jams the real NAP_{DEST_k} and impersonates NAP_{DEST_k} and EIPE_{SRC_k} when MD tries to associate.

**A*-mob-10** An attacker recovers K_j using BAM*-6 before handover. He then intercepts the control traffic exchanged between MD and SRC_k. On intercepting a handover-indication message sent from MD to HCN over SRC_k (over DEST_k) indicating handover to NAP_{DEST_k}, he uses AM*-2 to recover EK_k and IK_k and proceeds in the same way as in A*-mob-9.

**A*-mob-11** An attacker bids down the em_k, im_k negotiation (AM*-7) until em_k and im_k are disabled. He then intercepts the control traffic exchanged between MD and SRC_k. On intercepting a handover-indicating message sent from MD to HCN over SRC_k (over DEST_k) indicating handover to NAP_{DEST_k}, he jams the real NAP_{DEST_k} and impersonates NAP_{DEST_k} to MD.

**A*-mob-12** An attacker waits until HCN transfers the security context to DEST_k and then recovers IK_k and EK_k by AM*-3. He then proceeds as in A*-mob-9. Note that in A*-mob-12 the attacker has to get knowledge of DEST_k without having access to the data-protection keys. This is possible if MD does not encrypt control traffic (in this case, the attacker can intercept the plaintext of the handover indication message). If MD encrypts control traffic, the attacker can try to guess DEST_k from MD’s location.

**A*-mob-26** An attacker can try to jam the association between MD and DEST_k, he can try to make MD detect a handover reason although there is none, he can jam the
security-suite negotiation between MD and SRC\(_k\) or MD and DEST\(_k\), or he can bid down the negotiation to fail.

**A*-mob-27** An attacker can try to send a fake handover indication to SRC\(_k\) in order to keep SRC\(_k\) or HCN busy.\(^5\) An attacker can engage DEST\(_k\) in a fake security-mechanism negotiation or send fake handover request messages to DEST\(_k\) in order to keep DEST\(_k\) or HCN busy.\(^6\)

**A*-mob-29** An attacker can try to make MD discover a handover reason although there is none. In order to achieve this, the attacker can impersonate the network access point of a destination network with which HCN has a handover agreement, or amplify the signal of an existing network access point.

The attacks A*-14 to A*-16 cannot be adapted to the mobile-initiated case, as they make use of a fake handover-command message, which is specific to a network-initiated handover procedure. Consequently, in the mobile-initiated case, an attacker cannot simulate a handover procedure to MD on behalf of HCN and thus cannot violate the confidentiality in the root attack scenario RAS*-5.

However, in the mobile-initiated case, an attacker can try to simulate a handover procedure to HCN on behalf of a victim MD. We therefore consider the following additional root attack scenario:

**RAS*-12** An attacker tries to gain access to DEST\(_k\) by simulating a handover from SRC\(_k\) to DEST\(_k\).

An attacker can try conduct service theft in this additional root attack scenario by one of the following three attacks.

**A*-mob-14** An attacker recovers EK\(_{k-1}\) and IK\(_{k-1}\) with the help of BAM*-5 and uses them to send a fake handover indication message on behalf of MD to HCN over SRC\(_k\) (over DEST\(_k\)). Additionally, he uses AM*-1 to recover EK\(_k\) and IK\(_k\) according to the DEST\(_k\) identity he included in the fake handover-indication (handover-request) message. He then impersonates NAP\(_{DEST_k}\) and EIPE\(_{SRC_k}\) to MD when it tries to associate to it.

**A*-mob-15** An attacker recovers EK\(_{k-1}\) and IK\(_{k-1}\) with the help of AM*-6 or AM*-2 or AM*-1. He then uses EK\(_{k-1}\) and IK\(_{k-1}\) to send a fake handover-indication message to SRC\(_k\) on behalf of MD (over DEST\(_k\)) and uses AM*-2 to generate EK\(_k\) and IK\(_k\) according to the NAP\(_{DEST_k}\) identity he included in the fake handover-indication (request) message. He then proceeds as in A*-mob-14.

**A*-mob-16** An attacker disables em\(_{k-1}\) and im\(_{k-1}\) and sends a fake handover indication message on behalf of MD to SRC\(_k\) (over DEST\(_k\)). He uses AM*-2 to recover K\(_j\) and generate EK\(_k\) and IK\(_k\). He then proceeds as in A*-mob-14.

\(^{5}\)This is specific to a mobile-initiated procedure in which HCN is notified over SRC\(_k\).

\(^{6}\)This is specific to a mobile-initiated procedure in which HCN is notified over DEST\(_k\).
### Table 4.3: Overview on Attacks and Attack Modules

<table>
<thead>
<tr>
<th>Attack Module</th>
<th>Used in Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM*-1</td>
<td>A*-7, A*-9, A*-mob-9, A*-14, A*-mob-14, A*-15, A*-18, A*-20, A*-23, A*-25</td>
</tr>
<tr>
<td>AM*-3</td>
<td>A*-12, A*-mob-12</td>
</tr>
<tr>
<td>AM*-4</td>
<td>A*-1</td>
</tr>
<tr>
<td>AM*-5</td>
<td>A*-2, A*-3, A*-20</td>
</tr>
<tr>
<td>AM*-5 for 0 ≤ i ≤ k − 1</td>
<td>A*-5, A*-15, A*-mob-15, A*-20</td>
</tr>
<tr>
<td>BAM*-5 (0 ≤ i ≤ k − 1)</td>
<td>A*-5, A*-9, A*-mob-9, A*-14, A*-mob-14, A*-20</td>
</tr>
<tr>
<td>BAM*-6</td>
<td>A*-10, A*-mob-10, A*-17</td>
</tr>
<tr>
<td>BAM*-5 (i = k)</td>
<td>A*-8</td>
</tr>
<tr>
<td>AM*-7</td>
<td>A*-8, A*-11, A*-mob-11, A*-19, A*-22</td>
</tr>
<tr>
<td>Attacker disables $em_{k-1}$, $im_{k-1}$</td>
<td>A*-4, A*-21, A*-24</td>
</tr>
<tr>
<td>Attacker mounts MiM against $ka_0$</td>
<td>A*-13</td>
</tr>
<tr>
<td>Attacker breaks $a_0$</td>
<td>A*-13</td>
</tr>
<tr>
<td>Attacker fakes handover request</td>
<td>A*-27 A*-mob-29</td>
</tr>
<tr>
<td>Attacker fakes handover command</td>
<td>A*-14, A*-15, A*-16, A*-26</td>
</tr>
<tr>
<td>Attacker fakes handover indication</td>
<td>A*-mob-14, A*-15, A*-mob-16</td>
</tr>
</tbody>
</table>

Table 4.3 illustrates which attacks against a network-initiated or mobile-initiated procedure use which attack modules.

### 4.3.5 Requirements and Recommendations to Enhance SCT with Key Derivation

In this section, we generalize the requirements defined for a first-order handover with HN as anchor to the general case of a $k$-th-order handover with FN or HN as anchor. We differentiate between SRC-controlled, HN-controlled, and AN-controlled handover only if necessary.

Requirements R*-1 to R*-4 address security problems that arise from the use of weak security mechanisms between MD and any previously serving network SRC$_j$ ($1 ≤ j ≤ k$). Requirements R*-5 and R*-6 address problems arising from the use of weak mechanisms after the $k$-th-order handover. The requirements R*-7 and R*-8 protect the handover procedure messages themselves and prevent attacks arising from fake handover requests and commands.

In addition to the security requirements for a $k$-th-order handover procedure we, as in the first-order case, define recommendations (RC*-1 and RC*-2) on how DEST$_k$ and HCN...
should decide on whether or not to enter a handover agreement with each other. These recommendations cannot be addressed by a handover procedure alone. Protection against DoS attacks (like, for example, A*-26 to A*-27) is out of the scope of this work.

We first define and motivate the new requirements and then show how they address each of the attacks identified in the previous section.

Any of the key-establishment protocols $ke_j$ ($0 \leq j \leq k - 1$) used to establish data-protection keys between MD and SRC$_j$ may be vulnerable to the basic attack module BAM*-$1$ and thereby contribute to enable, for example, A*-7. Any of the previously used encryption and integrity-protection mechanisms may be partially or totally breakable and therefore vulnerable to the basic attack module BAM*-$5$. This module is used in the attacks A*-7, A*-9, A*-mob-9 A*-14, and A*-mob-14. The attacks A*-4, A*-16, and A*-mob-16 require $em_{k-1}$ and $im_{k-1}$ to be disabled. If any of the previously used key-derivation functions $kd_j$ ($0 \leq j \leq k - 1$) are vulnerable to the basic attack module BAM*-$2$ or BAM*-$3$, this enables the attacks A*-1 to A*-3, A*-6, A*-7, A*-9, A*-14, A*-15, A*-17, A*-18, A*-20, A*-23, and A*-25.

A broken initial authentication can be used to mount A*-13. A*-13 additionally requires that the initial key-agreement protocol $ka_0$ is vulnerable to a man-in-the-middle attack. It is therefore crucial for DEST$_k$ to be able to base its decision on whether to accept or refuse a given handover request on the initial security suite $(r)ss_0$, the cipher suites used so far, and the key-derivation function that was used to derive $kd_0$. Consequently, we require:

**R*-1** DEST$_k$ shall base its decision on whether to accept or refuse a given handover request on (1) the cipher suites $cs_0 = (ke_0, em_0, im_0), \ldots, cs_{k-1} = (ke_{k-1}, em_{k-1}, im_{k-1})$ previously used between MD and any of the previously serving networks; (2) the initial roaming authentication protocol $(r)a_0$; (3) the initial roaming key-agreement protocol $(r)ka_0$; and (4) the key-derivation method that was used to derive the transferred master session key $kd_k$.

This, for example, allows DEST$_k$ to refuse a handover command if no encryption and integrity protection was used between SRC$_k$ and MD.

In order to protect previous source networks and HCN from the use of a key-derivation function that is vulnerable to BAM*-$3$, we require:

**R*-2** Knowledge of a master session key $K_k$ (used by MD and DEST$_k$ after handover) shall not reveal any information on any previously used master session key $K_j$ ($0 \leq j \leq k - 1$).

This protects, for example, against the attacks A*-1, A*-2, A*-3, and A*-20.

In order to protect DEST$_k$ from the use of a key-derivation function that is vulnerable to BAM*-$2$, we require:

**R*-3** Knowledge of a previously used master session key $K_j$ ($0 \leq j \leq k - 1$) shall not reveal any information on $K_k$ to anyone except MD and DEST$_k$.

This protects, for example, against the attacks A*-6, A*-7, A*-9, A*-15, A*-17, A*-18, A*-20, A*-23, and A*-25.
Requirements R*-2 and R*-3 define the properties a key-derivation function should have, while R*-1 allows DEST\(_k\) to refuse handover requests if MD and HCN used a key-derivation function DEST\(_k\) is suspicious about.

**R*-4** The security-context transfer shall include information on the lifetime of the initial master session key \(K_0\).

This, for example, allows DEST\(_k\) to have considerations on the key lifetime as part of its handover decision. It furthermore allows HCN to pass on its restrictions on key lifetimes to DEST\(_k\).

It is crucial for HCN to influence the selection of the cipher suite to be used by MD and DEST\(_k\) after handover, as otherwise it may suffer from impersonation attacks (see A*-11, A*-22) due to the use of weak encryption and integrity-protection mechanisms \(em_1\) and \(im_1\). The use of a weak \(ke_1\) may lead to A*-12. We therefore require:

**R*-5** The negotiation of the cipher suite \(cs_k = (em_k, im_k, ke_k)\) (and optionally \(kd_k\)) shall enforce compliance with policies set by HCN, MD, and DEST\(_k\).

We identified bidding-down attacks against the negotiation of \(cs_k\) as one of the basic attack modules. This module is required in the attacks A*-8, A*-11, A*-19, and A*-22. In order to protect against these attacks, we require:

**R*-6** The negotiation of the cipher suite \(cs_k\) (and optionally \(kd_k\)) shall be protected against bidding-down attacks.

In order to protect DEST\(_k\) from fake handover requests as used in A*-27 and from recovery of the transferred key \(K_k\) during transfer (BAM*-6 (for \(i = k\)) used in A*-2 to A*-4, A*-12, and A*-20) and in order to allow DEST\(_k\) to verify HCN’s authorization of the handover, we require:

**R*-7** The security-context transfer between HCN and DEST\(_k\) on a \(k\)-th-order handover shall be encrypted and integrity-protected (including replay protection) and shall provide a proof of its origin (HCN) to DEST\(_k\).

In order to protect MD from fake handover-command messages (in the network-initiated case) as used in A*-14, A*-15, A*-16, or A*-26 and protect HN from fake handover-indication messages (in the mobile-initiated case) as required for the attacks A*-mob-14, A*-mob-15, A*-mob-16, and A*-mob-26 we require the handover command or the handover-indication message to be integrity-protected.

**R*-8** In the network-initiated case, the handover-command message shall be integrity-protected (including replay protection) and provide a proof of its origin (HCN) to MD. In the mobile-initiated case, the handover-indication message sent from MD to HCN over SRC\(_k\) (over DEST\(_k\)) shall be integrity-protected (including replay protection) and provide a proof of its origin (MD) to HCN.
DEST\textsubscript{k} may suffer from attacks that make use of the basic attack module BAM\textsuperscript{*}-6 and (c) of BAM\textsuperscript{*}-5. These basic attack modules make use of vulnerabilities of the network components and unprotected communication between network components in previously serving networks or HCN. A handover procedure cannot protect against these basic attack modules themselves. However, a key-derivation method that meets R\textsuperscript{*}-3 prevents attacks that use BAM\textsuperscript{*}-6 of BAM\textsuperscript{*}-5 as module in combination with AM\textsuperscript{*}-2. The only identified attack that uses BAM\textsuperscript{*}-6 or BAM\textsuperscript{*}-5 directly is A\textsuperscript{*}-5, from which SRC\textsubscript{j} (1 \leq j \leq k) but not DEST\textsubscript{k} suffers. With an eye on the potentially evolving attacks, we recommend:

**RC\textsuperscript{*}-1** A destination network shall carefully consider entering a handover agreement with HCN if HCN does not protect the memory of its AS\textsubscript{HCN} against unauthorized access (including physical access). DEST shall also carefully consider entering a handover agreement with HCN if other handover partners of HCN do not protect the memory of their authentication servers and encryption and integrity-protection endpoints against unauthorized access or do not protect the confidentiality and integrity of key transfers between their network components.

In particular, DEST\textsubscript{k} should not enter such handover agreements if R\textsuperscript{*}-3 is not met by the key-derivation method applied.

**RC\textsuperscript{*}-2** A HCN shall carefully consider entering into handover agreements with destination networks that do not protect the memory of their EIPE\textsubscript{DEST} and/or AS\textsubscript{DEST} against unauthorized access (including physical access) that do not protect the integrity and confidentiality of key transfers from AS\textsubscript{DEST} to EIPE\textsubscript{DEST}, or that keep K in EIPE\textsubscript{DEST}'s memory longer than necessary.

In particular, HCN shall not enter agreements with DESTs of this kind if it could be held responsible for the impersonation attack A\textsuperscript{*}-12. Note that other attacks with impact of HCN that make use of any of the above vulnerabilities in DEST\textsubscript{k} can all be prevented by the use of a key-derivation function that meets R\textsuperscript{*}-2.

In order to avoid an attacker to exploit a source network’s channel reservation for a denial of service attack by suppressing the handover-complete message, we recommend that fall-back times on hard handover procedures be carefully set by all source networks. This, and other ways to thwart DoS attacks (A\textsuperscript{*}-26 to A\textsuperscript{*}-27), is, however, out of scope of this work.

A subsequent handover procedure that meets the above requirements is protected against the attacks A\textsuperscript{*}-1 to A\textsuperscript{*}-25 (network-initiated case) or A\textsuperscript{*}-1 to A\textsuperscript{*}-8, A\textsuperscript{*}-mob-9 to A\textsuperscript{*}-mob-16, and A\textsuperscript{*}-17 to A\textsuperscript{*}-25 (mobile-initiated case). As illustrated in Table 4.3, each of these attacks makes use of one of the attack modules listed on the left side of the table. Consequently, it is sufficient to protect against these attack modules. Table 4.4 shows how the above-defined requirements address the attack modules listed on the left side of Table 4.3.

\textsuperscript{8}In all of the identified attacks, BAM\textsuperscript{*}-4 is used in combination with one of the other attack modules.
### 4.3 $k$-th-order Handover—SCT with Key Derivation

<table>
<thead>
<tr>
<th>Attack Module</th>
<th>Adressed by</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAM*-1 ($0 \leq i \leq k - 1$) (AM*-1)</td>
<td>R*-1 DEST$_k$ bases its decision on all $ke_j$ ($0 \leq j \leq k - 1$)</td>
</tr>
<tr>
<td>BAM*-2 (AM*-1, AM*-2)</td>
<td>R*-3</td>
</tr>
<tr>
<td>BAM*-4 for ($i = k$)</td>
<td>not addressed$^8$</td>
</tr>
<tr>
<td>BAM*-1 for ($== k$) (AM*-4)</td>
<td>R*-5 policies of HCN during $ke_k$ negotiation enforced</td>
</tr>
<tr>
<td>BAM*-3 (AM*-4, AM*-5)</td>
<td>R*-2</td>
</tr>
<tr>
<td>BAM*-5 (AM*-1)</td>
<td>R*-1 DEST bases its decision on $em_j$, $im_j$ ($0 \leq j \leq k - 1$)</td>
</tr>
<tr>
<td>BAM*-6 for $0 \leq i \leq k - 1$ (AM*-2, AM*-6)</td>
<td>RC*-1</td>
</tr>
<tr>
<td>BAM*-6 for $i = k$ (AM*-3, AM*-5)</td>
<td>RC*-2 and R*-7</td>
</tr>
<tr>
<td>BAM*-5 for $i = k$ (AM*-4)</td>
<td>R*-5 policies of HCN during $em_1$ and $im_1$ negotiation enforced</td>
</tr>
<tr>
<td>AM*-7</td>
<td>R*-6</td>
</tr>
<tr>
<td>Attacker disables $em_{k-1}$, $im_{k-1}$</td>
<td>R*-1</td>
</tr>
<tr>
<td>Attacker breaks $a_0$</td>
<td>R*-1</td>
</tr>
<tr>
<td>Attacker mounts MiM against $ka_0$</td>
<td>R*-1</td>
</tr>
<tr>
<td>Attacker fakes handover command</td>
<td>R*-8 and R*-1</td>
</tr>
<tr>
<td>Attacker fakes handover request</td>
<td>R*-7 and R*-1</td>
</tr>
<tr>
<td>Attacker fakes handover initiation</td>
<td>R*-8 and R*-1</td>
</tr>
</tbody>
</table>

Table 4.4: Basic Attack Modules and Requirements
4.4 Differences on SCT with Key Agreement

Most naturally, the root attack scenarios RAS*-1 to RAS*-10 are the same in both the case of SCT with key agreement and the case of SCT with key derivation. However, the key agreement protects against some of the potential attacks that occur in SCT with key derivation.

In particular, the basic attack modules BAM*-1, BAM*-4, and BAM*-5 are exactly the same for SCT with key agreement as for SCT with key derivation. However, BAM*-6 differs in case of SRC-controlled handover with a key agreement that generates multiple initial keys: in this case, $K_i$ can be recovered from any previous security-context transfer $S_j$, $(1 \leq j \leq i - 1)$ as each of them includes the keys $K_j \ldots K_n$. In case HCN is HN or AN, BAM*-6 is the same as in the case of SCT with key derivation.

The basic attack modules BAM*-2 and BAM*-3 are specific to the key-derivation case. In the case of key agreement, no key relations between previously used and future master session keys can be exploited by an attacker. As a consequence, only the attack modules AM*-3, AM*-6, and AM*-7 can be exploited by an attacker in the case of SCT with key agreement.

These attack modules can be used as modules for A*-4, A*-8, A*-11 (A*-mob-11), A*-12 (A*-mob-12), A*-19, A*-21, A*-22, A*-24, A*-26 (A*-mob-26), A*-28, A*-29 (A*-mob-27), and A*-27 (A*-mob-29). The attacks A*-5 and A*-20 can be used in connection with BAM*-4. All of the other attacks described in Section 4.3 exploit the relation between the master keys and are therefore specific to SCT with key derivation.

In Section 3.2.4, we introduced two possible ways to agree upon fresh keys to transfer in the security context. One way is to execute a new run of a roaming key-agreement protocol via SRC_k before handover. The second possibility is to generate multiple keys during the initial roaming authentication and key agreement between MD and AN. However, as any key-agreement protocol that derives a fresh master session key $K_k$ based on the credentials exchanged between MD and HN could be used on this type of SCT, we describe the requirements more generally.

Due to using a key agreement, the requirements R*-2, R*-3 and R*-4 become obsolete. Instead, we introduce the new requirement R’-2. The requirements R*-5, R*-6, and R*-8, as well as the recommendations RC*-1 and RC*-2, defined for subsequent handover, however, have to be met by SCT with key agreement. The requirements R*-1 and R*-7 can be somewhat relaxed:

**R’-1** DEST_k shall base its decision on whether to accept or refuse a given handover request on (1) the last authentication protocol used between MD and any previously serving network; (2) the key-agreement protocol used to agree upon the master session key $K_k$; and (3) the cipher suite used between MD and HCN during the negotiation of $c_{sk}$.

The use of a weak cipher suite between HCN and MD during the negotiation of $cs_k$ may be exploited for bidding-down attacks. Weaknesses in any other cipher suites used between MD and any previously serving networks are without consequence for DEST_k as MD and DEST_k use a fresh master key.
4.5 Related Work

R'-2 The key-agreement protocol used to agree upon the master session key $K_k$ shall not reveal any information on $K_k$ to anyone but MD and DEST$_k$.

R'-7 The security-context transfer between HCN and DEST$_k$ on a $k$-th-order handover shall be integrity-protected (including replay protection) and shall provide a proof of its origin (HCN) to DEST$_k$. Furthermore, the security-context transfer shall be encrypted in case it includes any confidential information.

In the two key-agreement methods described previously (see Section 3.2.4) the security context includes confidential information. However, in Section 5.3, we introduce a key-agreement method that does not require any secret information to be transferred from HCN to DEST$_k$.

A handover procedure that uses SCT with key agreement and meets the here-defined requirements is protected against all attacks (except the DoS attacks) identified for this type of SCT. The requirements address the attacks in the same way as in the SCT with key-derivation case.

4.5 Related Work

SCT has recently been discussed in the context of first-order inter-provider handover [74, 75, 111, 162, 176, 177, 186].

This previous work defines security requirements equivalent to R-2, R-8, and R-7, but it is restricted to first-order handover only. Similarly, [176, 177] consider R-3 for first-order handover only. In contrast, the first-order requirements R-1, R-6, R-5, or R-4 are neither defined nor addressed in previous work. The authors of [74, 75, 111, 176] concentrate on how SCT potentially allows for faster inter-provider handover, but they address only R-7 and R-8. It is only in [162, 177, 186] that the first-order variant of R-2 is addressed and only Wang et al. [177] also address R-3 in the first-order variant.

Soltwisch et al. [162] suggest deriving $K_0$ from $K_1$ by adding a random number $r_1$ to $K_0$. Upon transfer to MD, $r_1$ is integrity-protected but sent in the clear. By construction, SRC (DEST) and any attacker that obtained knowledge of $K_0$ ($K_1$) can thus easily obtain $K_1$ ($K_0$) by intercepting $r_1$. Consequently, this key-derivation method neither meets R-2 nor R-3.

Zhang et al. [186] suggest deriving $K_1$ by means of a pseudo-random function with $K_0$ and a random number $r_1$ as input. As in [162], $r_1$ is transferred to MD in the clear. However, the use of the pseudo-random function as a key-derivation function guarantees that R-2 is partially met. R-3 is not addressed.

Wang et al. [177] suggest deriving $K_1$ in the same way as in [162]. However, the currently serving network transfers $K_1$ to MD encrypted with an encryption key $EK_0$ shared between SRC and MD. This key-derivation method meets R-2. However, an attacker that gained knowledge of $EK_0$ can intercept and decrypt the key transfer and thus obtain $K_1$. It is, in our opinion, not a good solution to transfer the future master key to MD protected by
the old data-protection keys. Furthermore, by construction, the source network in their handover procedure gains knowledge of $K_1$. In order to meet R-3, Wang et al. suggest using the transferred master key $K_1$ to authenticate a Diffie-Hellman key-exchange between MD and DEST after handover and derive data-protection keys for use after handover from the exchanged key. However, the authors fail to notice that by knowledge of $K_0$, the source network (or any attacker with knowledge of $K_0$) can mount a man-in-the-middle attack against the key-exchange. Consequently, R-3 cannot be met by this method.

4.6 Conclusion

In this chapter, we have presented a threat analysis for all modeled SCT types and all modeled handover procedures and defined new security requirements. A handover procedure that meets these requirements is protected against all but the Denial of Service (DoS) attacks identified in the threat analysis.

The need for a thorough threat analysis for SCT on handover was previously stated in [75, 111]. First steps to specify security requirements for SCT on handover were taken in [162, 176, 177]. Our work exceeds this previous work in (1) explicitly modeling threats arising from subsequent context transfers, i.e., modeling the impact of weak security mechanisms used between MD and any previously serving network; (2) describing concrete attack scenarios for first-order as well as higher-order handover using our security model for wireless networks; (3) exploring the impact of the key relationships arising from SCT with key derivation; (4) recognizing the importance of a cipher-suite negotiation that enforces compliance with policies set by HCN, as well as MD and DEST; and (5) the protection of this cipher-suite negotiation against bidding down.

Although in our threat analysis we consider DoS attacks, we do not make any suggestions on how to prevent these attacks. This is an interesting topic for future research.
Chapter 5

New History-Enriched Policy-Based SCT for Inter-Provider Handover

In this chapter, we present network-initiated and mobile-initiated handover procedures with SCT and key derivation for all handover control types that meet the requirements R*-1 to R*-8 defined in Section 4.3, except for R*-3 which is only met in part [124]. Moreover, we present handover procedures with SCT and key agreement that meet all the requirements defined in Section 4.4. The new components of these procedures are a context history, key-derivation and key-agreement methods, as well as a specification of policies and handover agreements. The handover procedures themselves include various new methods to negotiate the cipher suite to be used after handover.

The context history in our new approach not only protects all participants in a handover procedure from attacks arising from the use of any sort of weak mechanisms before this handover, but it also enables the inter-operation of providers supporting different security levels.

The security context is enriched with a context history and a threshold on the lifetime of the transferred master session key set by HCN. The context history includes information on the initial security suite. In the case of SCT with key derivation, the history additionally includes information on all previously used cipher suites, as well as the lifetime of the initial master session key (R*-4). The context history enables DEST_k to base its decision on whether to accept or refuse a handover request on previously used security mechanisms (R*-1, R*'-1).

Furthermore, in our approach DEST_k, HCN and MD set handover policies on the cipher suites they allow to be used after handover dependent on the context history. We present various methods to negotiate the cipher suite to be used by MD and DEST_k after handover. All of these methods enforce the choice of the cipher suite to comply with policies set by HCN, MD, and DEST_k, and consequently meet R*-5. This protects against attacks arising from the use of weak security mechanisms after handover. Furthermore, we show how each
of these negotiation methods can be protected against bidding-down attacks (R*-6).

In our network-initiated handover procedures, the handover-command message sent from HCN to MD is integrity-protected and in our mobile-initiated handover procedures, the handover-indication message is integrity-protected (R*-8). The security-context transfer from HCN to DEST_k is sent over an encrypted and authenticated channel (R*-7). In the case of SCT with key derivation, the key-derivation functions used fully meet R*-2 and meet R*-3 in part.

The HEPB-based approach for SCT with key-derivation is joint work with S. Wetzel and will partly be published in [124].

Moreover, we show that the two key-agreement methods for SCT discussed so far (see Section 3.2.4) meet R’-2 only in part. Therefore, we introduce a new key-agreement method by means of which a master session key can be agreed upon between MD and DEST_k during HN-controlled handover. This key-agreement method does not even reveal any information on K_k to HN and thus meets R’-2. The method makes use of a secret-sharing approach similar to the one introduced in Section 2.2. As the fresh master key does not have to be transferred in the clear from HCN to DEST_k, this approach does not require the authenticated channel between DEST_k and HCN to be encrypted.

Outline. In Section 5.1, we present the history-enriched, policy-based (HEPB) handover approach for SCT with key derivation. We start by specifying a network-initiated first-order handover procedure with HN as anchor in Section 5.1.1. We then generalize this procedure to the subsequent handover case in Section 5.1.2. The differences in case of a mobile-initiated subsequent HEPB handover procedure are summarized in Section 5.1.3. In Section 5.2, we present the HEPB handover approach for SCT with key agreement. In Section 5.3, we show how a secret-sharing approach can be used to establish fresh keys between MD and DEST_k. Section 5.4 relates our new HEPB procedures to previous work in this field and concludes the chapter.

5.1 HEPB SCT with Key Derivation

In this section, we introduce the history-enriched, policy-based approach for SCT with key derivation. We start by describing a network-initiated first-order handover procedure with HN as anchor. We then generalize this first-order procedure to a k-th-order handover procedure with HN or FN as anchor (including first-order handover with HN or FN as anchor). Finally, we describe the differences in case of a mobile-initiated k-th-order handover procedure.

5.1.1 First-Order Network-Initiated HEPB Handover with HN as Anchor

In this section, we present the history-enriched, policy based-handover procedure for a first-order handover with HN as anchor.
5.1.1.1 Context History

In order to allow for DEST to base its decision on whether to accept or refuse a handover request on the security suite used between MD and HN before handover (R-1), we include

$$\text{history}_0 = (ss_0, kd_0, T_0), \quad \text{where} \quad ss_0 = (a_0, ka_0, ke_0, em_0, im_0)$$

in the security context $S_1$ transferred from HN to DEST during a first-order handover.\(^1\) Here, $ss_0$ is the initial security suite and $kd_0$ is the key-derivation function used to derive $K_1$ from $K_0$. We refer to $(ss_0, kd_0)$ as the security-suite history $ssh_0$. HN also adds a lifetime indicator $T_0$ to the context to provide information on the total lifetime of the initial security context at the time of initiation of the first-order handover (R-4).\(^2\)

5.1.1.2 Security Policies

MD, HN, and any destination network DEST have policies with respect to which cipher suites they allow to be used after a first-order handover, given a particular security-suite history $ssh_0$. They express these policies by pre-defining sets $CS_{MD}|_{ssh_0}$, $CS_{HN}|_{ssh_0}$, and $CS_{DEST}|_{ssh_0}$ of cipher suites $cs = (ke, em, im)$ for any possible security-suite history $ssh_0$. MD, HN, and DEST set $CS_{MD}|_{ssh_0}$, $CS_{HN}|_{ssh_0}$, or $CS_{DEST}|_{ssh_0}$ to be empty if and only if they do not allow for handover for a particular security-suite history $ssh_0$ at all.\(^3\)

As a simple example, MD may have the policy that the cipher suite used before handover must be used after handover as well. MD can express this policy by setting $CS_{MD}|_{ssh_0} = \{(ke_0, em_0, im_0)\}$.

Furthermore, MD, HN, and DEST each have a policy setting an upper boundary on how long an initial security context may be used. To express this policy MD, HN, and DEST each define a threshold $Tr_{MD}$, $Tr_{HN}$, and $Tr_{DEST}$.

5.1.1.3 Handover Agreement

HN enters first-order handover agreements with destination networks DEST. Such an agreement regulates terms and conditions (including, for example, accounting issues) for HN-controlled first-order handover to DEST. The handover agreement includes an exchange of credentials that allow HN and DEST to establish an authenticated and encrypted channel.

\(^1\)Here, we assume that $S_1$ by means of the security suite history implicitly carries the information on which technologies were used on before and after each previous handover. If this is not the case, history should additionally include $T$.

\(^2\)We do not further specify how $T_0$ is determined. It should, however, be a measure of the lifetime of the initial security context measured in both time units, as well as the amount of data that was so far encrypted and integrity-protected with keys derived from the initial master key $K_0$. Note that $T_0$ may consist of more than one component.

\(^3\)CS$_{MD}|_{ssh_0}$, CS$_{HN}|_{ssh_0}$, and CS$_{DEST}|_{ssh_0}$ are subsets of CS, the set of all cipher suites specified for the given wireless technology.
As part of the handover agreement, DEST commits to its handover policies by means of supersets $CS^*_{DEST}|_{ssh_0} \supset CS^*_{DEST}|_{ssh_0}$. By committing to $CS^*_{DEST}|_{ssh_0}$, DEST commits to refuse a handover request including a specific security-suite history $ssh_0$ in case no cipher suite $cs_1 \in CS^*_{DEST}|_{ssh_0}$ can be negotiated.

Committing to a superset rather than $CS_{DEST}|_{ssh_0}$ for each security-suite history $ssh_0$ provides DEST with some degree of flexibility. It allows DEST to further restrict its policy over time (i.e., exclude certain cipher suites from $CS_{DEST}|_{ssh_0}$ or set $CS_{DEST}|_{ssh_0}$ to be empty for a particular security-suite history $ssh_0$) without prior notice to HN. By committing to $CS^*_{DEST}|_{ssh_0}$, DEST, however, explicitly excludes specific security suite histories after which it will not allow handover at all, already at the time of entering the handover agreement. In turn, DEST’s commitment allows HN to pre-select candidate networks and consequently avoid requesting handover to a destination network in vain.

If DEST does in fact not want to make any commitment, but rather wants to stay as flexible as possible, DEST can simply commit to $CS^*_{DEST}|_{ssh_0} = CS$. If DEST does not accept handover for a particular security suite $ss_0$, it commits to $CS^*_{DEST}|_{ssh_0} = \emptyset$ for this security suite.

Similarly, DEST commits to an upper boundary $u_{DEST}$ on $Tr_{DEST}$. By committing to $u_{DEST}$, DEST commits to refusing a handover request if the lifetime indicator $T_0$ included in the context history $history_0$ exceeds $u_{DEST}$. This commitment again helps HN to pre-select candidate destination networks and allows DEST to lower its threshold without prior notice to HN.

5.1.1.4 Key Derivation and Security Context

In order to meet R-2, we require $kd_0$ to be a pre-image resistant hash function (see [120] for definition) that takes $K_1$, the identity of DEST, and a random number $RAND$ as input.\(^4\) The pre-image resistance guarantees that knowledge of $K_1$ does not reveal any information on $K_0$. However, knowledge of $K_0$ and DEST’s identity implies knowledge of $K_1$ such that R-3 cannot be met by this key-derivation method.

HN includes the previously defined context history $history_0$, as well as its threshold $Tr_{HN}$\(^5\) in the security context $S_1$ such that

$$S_1 = (K_1, Tr_{HN}, history_0).$$

5.1.1.5 Network-Initiated First-Order Procedure

In this section, we describe the history-enriched, policy-based handover procedure itself. We reuse the procedure model illustrated in Figure 3.12. We will discuss various methods

---

\(^4\)DEST’s identity guarantees that HN derives different keys for different destination networks. This is important for subsequent handover. $RAND$ in addition guarantees that the master key $MD$ is handed back to a previous source network at some point in a chain of subsequent handover, the new master key to be used cannot be pre-determined by the previous source network.

\(^5\)Note that it is not sufficient to add $Tr_{HN} - T_0$ to the security context, as the respective threshold $Tr_{DEST}$ of DEST may be smaller than $Tr_{HN}$. 

5.1 HEPB SCT with Key Derivation

to negotiate the cipher suite $cs_1$ at the end of this section. In the following description of the procedure, we only describe their common interfaces.

Before handover, MD is connected to HN. MD and HN use the initial encryption mechanism $em_0$ and the initial integrity-protection mechanism $im_0$ to protect data and control traffic between MD and EIPE$_{HN}$.

In the security-mechanisms negotiation phase (1), MD and HN negotiate on the cipher suite to use after a first-order handover. This negotiation phase takes $CS_{MD}|_{ssh_0}$ and $CS_{HN}|_{ssh_0}$ as input and outputs a subset of their intersection

$$\text{Nego}_1 \subset CS_{MD}|_{ssh_0} \cap CS_{HN}|_{ssh_0}.$$ 

Upon detecting a reason for inter-provider handover, HN determines an ordered list $L = \{\text{DEST}^1, \ldots, \text{DEST}^n\}$ of candidate destination networks. We do not further specify how HN determines $L$. However, we, without loss of generality, assume that HN will ensure that $L$ only contains destination networks that have a handover agreement with HN.

HN picks the first candidate network $\text{DEST}^1$ in $L$ and checks whether $T_0 < u_{\text{DEST}^1}$. HN then checks whether $CS_{\text{DEST}^1}|_{ssh_0} \cap \text{Nego}_1$ is not empty. If either of these checks fail, HN restarts the selection process with $\text{DEST}^2$. If both checks are successful, HN sends a handover request to $\text{DEST}^1$. HN proceeds like this until it sends a handover request to some destination network $\text{DEST}^i$ ($1 \leq i \leq n$) or reaches the end of the candidate list $L$. In the latter case, no handover is possible and HN drops the connection to MD. Figure 5.1 illustrates the selection procedure executed by HN.

![Figure 5.1: Details of “select DEST” of Figure 3.12](image)

The handover request sent from HN to $\text{DEST}^i$ is authenticated and integrity-protected by means of keys agreed upon between HN and $\text{DEST}^i$ based on the credentials established on entering the handover agreement (R-7). It includes the security context $S_1$ defined
above, the output \( \text{Nego}_1 \) of the first negotiation phase, as well as identities of MD, HN,\(^6\) and DEST:\(^6\)

\[
\text{handover request} = (\text{MD}_{ID}, \text{HN}_{ID}, \text{DEST}^i_{ID}, S_1).
\]

Upon receipt of the handover request, DEST\(^i\) decides whether to reject the handover request or to enter the security-mechanism negotiation phase (2) with HN. DEST\(^i\) first checks whether it received the handover request over an encrypted channel from an HN with which it has a handover agreement (see R-7). DEST then checks if \( T_0 < T_{\text{DEST}^i} \) and whether \( CS_{\text{DEST}^i_{ssl}} \cap \text{Nego}_1 \) is not empty (see R-1). If either of these checks fail, DEST\(^i\) returns a negative handover response.\(^7\) If all of these checks are successful, DEST\(^i\) sends a positive handover response back to HN. The decision process of DEST\(^i\) is illustrated in Figure 5.2.

![Figure 5.2: Details of “DEST\(^i\) decision” of Figure 3.12](image)

Upon receipt of a positive handover response, HN starts to negotiate a cipher suite \( cs_1 \) with DEST\(^i\) (security-mechanism negotiation (2) in Figure 3.12). This negotiation phase takes \( \text{Nego}_1 \) and \( CS_{\text{DEST}^i_{ssl}} \) as input and outputs a subset \( \text{Nego}_2 \) of the intersection of these two sets. If the negotiation is successful (i.e., \( \text{Nego}_2 \neq \emptyset \)), HN selects DEST\(^i\) as the next destination network DEST and sends a handover-command message to MD including the identity of DEST and the random number RAND. Upon receipt of a negative response or if the negotiation fails, HN sets \( i = i + 1 \) and restarts the selection process.

The handover-command message is integrity-protected with the integrity-protection key \( IK_0 \) and the integrity-protection mechanism \( irn_0 \) (see R-8). Upon receipt of a handover command message MD checks whether the current lifetime of the initial security context \( T \) exceeds \( T_{\text{RMD}} \). If this is the case, MD drops the connection. Otherwise MD extracts the

---

\(^6\)Depending on the identity type used, the identity of MD may already contain HN’s identity. In this case, the identity of HN does not have to be included again in the handover request.

\(^7\)As already noted, DEST\(^i\) may also reject a handover request for other reasons than security reasons, e.g., if it does not have the free capacity to serve MD after handover.
random number \( RAND \) and computes the master session key \( K_1 \) from DEST’s identity and \( RAND \) with the help of \( kd_0 \).

If \( \text{Nego}_2 \) consists of more than one element, MD and HN may negotiate which of them to use after handover in the security-mechanism negotiation phase (3). MD may then inform DEST of their negotiation result in the security-mechanism negotiation phase (4) during the association.

Security-Mechanism Negotiation

In Section 4.2.5, we require that on a first-order handover with HN as anchor, the negotiation of the cipher suite \( cs_1 \) shall enforce compliance with policies set by MD, DEST, and HN (see R-5). Moreover, the cipher-suite negotiation has to be protected against bidding-down attacks (see R-6). In this section, we present several different negotiation methods that meet R-5 and R-6 and show how these methods can be integrated into the history-enriched, policy-based first-order handover procedure.

We differentiate between two types of negotiation methods again: methods that ignore preferences of the negotiating parties and methods that respect these preferences.

**Negotiation without Preferences.** In this section, we describe two negotiation methods that do not take into account the preferences of MD, HN, and DEST\(^i\) on the cipher suites they allow to be used after handover. In Method 1, MD reveals its handover policies to HN during the pre-registration.\(^8\) In the second method, HN obtains the set of cipher suites MD allows to be used from MD while MD is associated with HN.

**Method 1.** For each security-suite history \( sssh_0 \), MD reveals the set of cipher suites \( CS_{MD}|_{ssh_0} \) to HN as part of the pre-registration process. HN looks up the sets \( CS_{MD}|_{ssh_0} \) and \( CS_{HN}|_{ssh_0} \) and computes their intersection \( \text{Nego}_1 \). HN then proceeds with the candidate destination networks selection as illustrated in Figure 5.1.

HN includes \( \text{Nego}_1 \) in the handover request it sends to DEST\(^i\). Upon receipt of the handover request, DEST\(^i\) computes \( \text{Nego}_2 = \text{Nego}_1 \cap CS_{DEST}|_{ssh_0} \). DEST\(^i\) sends a negative handover response to HN if \( \text{Nego}_2 = \emptyset \). Otherwise, DEST\(^i\) selects \( cs_1 \) from the cipher suites in \( \text{Nego}_2 \) and includes \( cs_1 \) in a positive handover response it sends back to HN.

HN checks whether \( cs_1 \) is an element of \( CS_{HN}|_{ssh_0} \). In the negative case, HN restarts the selection process with the next destination network in \( L \). In the positive case, HN sends a handover command, including DEST\(^i\)’s identity as well as \( cs_1 \) to MD. MD checks whether \( cs_1 \in CS_{MD}|_{ssh_0} \). In the negative case, MD sends a failure message back to HN.

This negotiation method takes policies set by HN, MD, and DEST into account. These policies are enforced as DEST selects \( cs_1 \), and MD and HN check whether \( cs_1 \) is indeed a cipher suite for which they allow (see R-5).

The handover request sent from HN to DEST\(^i\) that includes the set \( \text{Nego}_1 \) of cipher suites HN and MD allow to be used after handover is integrity-protected by means of keys agreed upon based on the credentials established between HN and DEST\(^i\) as part

---

\(^8\)MD may change this policies, e.g., through a web interface provided by its HN.
of entering their handover agreement. The handover-command message sent from HN to MD that includes the nego-
tiated cipher suite $cs_1$ is integrity-protected by means of the ini-
tial integrity-protection mechanism $im_0$ and the initial integrity key $IK_0$. The integrity 
protection of these two messages guarantee that this negotiation method meets R-6.

The disadvantage of this negotiation method is that HN has to store all handover policies 
for all security suite histories $ssh_0$ for each of its pre-registered users.

**Method 2.** To cut down the amount of data HN has to store, Method 1 can be changed 
in the following way: Instead of committing to its policies as part of the pre-registration, 
MD sends the set of cipher suites $CS_{MD}\{ssh_0\}$ it allows to be used after handover for the 
current initial security context to HN prior to a handover reason detection. The transfer of 
this message is integrity-protected by means of $im_0$ and $IK_0$. HN computes the intersection 
$Neg_1 = CS_{HN}\{ssh_0\} \cap CS_{HN}\{ssh_0\}$ and the following HN, DEST, and MD proceed as in 
Method 1.

Method 2 meets R-6 as the transfers of $CS_{MD}\{ssh_0\}$, $Neg_1$ and $cs_1$ are integrity-
protected. R-5 is met by Method 2 in the same way as in Method 1.

It is interesting to note that in both methods DEST can refuse handover if no or a weak 
integrity protection was used before handover by setting the set of cipher suites it allows 
to be used to be empty. DEST thus has additional control over the protection of the $cs_1$ 
negotiation against bidding down.

**Negotiation with Preferences.** In this section, we discuss some negotiation methods 
that take into account preferences of HN, MD, and DEST with respect to the cipher suites 
they allow to be used. We assume that HN, MD, and DEST each have a preference order 
$\leq_{HN}$, $\leq_{MD}$, and $\leq_{DEST}$ on each of their handover policies. These preference orders differ 
from each other as HN, MD, and DEST may assert the security level of cipher suites 
differently.

Negotiation mechanisms that respect the preferences of the negotiating parties can be 
symmetric and weigh the preferences of each party equally high. In many cases, however, 
an asymmetric negotiation mechanism that orders the negotiating parties and respects the 
preference of the party ranked highest over the other parties seems more suitable. Imagine, 
for example, that HN guarantees to reimburse DEST for service provisioning for its pre-
registered users. In this case, HN takes a bigger risk if DEST and MD use weak security 
mechanisms than DEST does. Thus HN’s preferences should outweigh DEST’s preferences.

In the remainder of this section, we discuss one example for a negotiation method that 
respects HN’s preferences highest, one that weighs DEST’s preferences highest, and one 
that weighs MD’s preferences highest. Method 3 presented in Section 1.3 that respects the 
preferences of two negotiating parties equally high cannot easily be adapted to the handover 
case, as it requires several round-trips between the negotiating parties.

**Method 3.** In this method, HN’s preferences weigh highest, MD’s preferences next, and 
DEST’s preferences weigh the lowest. MD sends the set of cipher suites $CS_{MD}\{ssh_0\}$, as well 
as its preference order $\leq_{MD}$ on $CS_{MD}\{ssh_0\}$, to HN in Phase (1) before a handover reason is 
detected. DEST sends $CS_{DEST}\{ssh_0\}$ and $\leq_{DEST}$ to HN in Phase (2) as part of its handover
response. HN then computes the intersection of \( CS_{MD|ssh_0}, CS_{HN|ssh_0} \) and \( CS_{DEST|ssh_0} \) and orders the resulting set of cipher suites according to its preference order \( \preceq_{HN} \). If HN prefers more than one cipher suite most, it orders these according to MD’s preferences. If MD prefers more than one of the remaining cipher suites most, HN orders these according to DEST’s preferences. From the remaining cipher suites, HN randomly picks \( cs_1 \):

\[
cs_1 \in_R \max_{\preceq_{DEST}} \left\{ \max_{\preceq_{MD}} \left\{ \max_{\preceq_{HN}} \{ CS_{HN|ssh_0} \cap CS_{MD|ssh_0} \cap CS_{DEST|ssh_0} \} \right\} \right\}
\]

HN sends its choice \( cs_1 \) back to DEST\(^i\) and DEST\(^i\) checks that \( cs_1 \) is indeed a cipher suite it allows to be used given the current security-suite history. HN additionally includes \( cs_1 \) in the handover command it sends to MD and MD checks whether \( cs_1 \in CS_{MD|ssh_0} \).

As MD and DEST check that \( cs_1 \) complies with their respective policy, Method 3 enforces compliance with their policies. HN, on the other hand, selects the cipher suite to be used and can therefore enforce its policy as well (see R-5).

The protection against bidding-down attacks (R-6) works similar to Method 2: MD protects the integrity of the transfer of \( CS_{MD|ssh_0} \) to HN by means of \( imt_0 \) and \( IK_0 \). DEST\(^i\) uses keys agreed upon based on the credentials established between DEST\(^i\) and HN as part of entering a handover agreement to integrity-protect the transfer of \( CS_{DEST|ssh_0} \) to HN.

**Method 4.** In this method, the preferences of DEST\(^i\) weigh highest, those of HN second, and those of MD lowest. MD again sends \( CS_{MD|ssh_0} \) to HN in Phase (1). HN computes the intersection \( Nego_1 = CS_{HN|ssh_0} \cap CS_{MD|ssh_0} \) and includes Nego_1 in the integrity-protected handover request sent to DEST\(^i\). DEST\(^i\) computes the intersection of Nego_1 with \( CS_{DEST|ssh_0} \) and orders the result Nego_2 according to its own preference order \( \preceq_{DEST} \). If more than one cipher suite is maximal according to the preference order of DEST\(^i\), it orders these according to the preferences of HN. If more than one cipher suite is maximal according to HN’s preferences, DEST\(^i\) orders these according to MD’s preferences. DEST\(^i\) then randomly picks a cipher suite from the remaining cipher suites:

\[
cs_1 \in_R \max_{\preceq_{MD}} \left\{ \max_{\preceq_{HN}} \left\{ \max_{\preceq_{DEST}} \{ Nego_2 \cap CS_{DEST|ssh_0} \} \right\} \right\}
\]

DEST\(^i\) includes its choice \( cs_1 \) in the handover response to HN and HN checks that \( cs_1 \in CS_{HN|ssh_0} \). HN informs MD of the choice of \( cs_1 \) in the handover command message (Phase (3)). MD checks that \( cs_1 \in CS_{MD|ssh_0} \).

The protection against bidding down (R-6) works as in Method 1. Moreover, R-5 is met by MD and HN checking the compliance of DEST\(^i\)’s choice with their policies.

**Method 5.** In this method, MD’s preferences weigh highest. DEST\(^i\) sends \( CS_{DEST|ssh_0} \) to HN as part of the handover response and protects the integrity of this message by means of keys agreed upon between DEST\(^i\) and HN based on the credentials exchanged on entering the handover agreement. HN computes the intersection of the received set of cipher suites and \( CS_{HN|ssh_0} \) and sends the result Nego_2 to MD as part of the integrity-protected handover.

\(^9\)Alternatively, DEST\(^i\) could inform MD of its choice during association. However, in the above method, HN can enforce its policy and MD gets a guarantee that HN approves of the choice of \( cs_1 \).
command. MD then proceeds analogous to DEST in the last method, replacing MD with DEST. MD informs HN of its choice \( cs_1 \) (Phase(3)). HN checks whether \( cs_1 \) complies with its policy. In the positive case, HN sends \( cs_1 \) in an integrity-protected message to DEST. DEST, in turn, checks that \( cs_1 \) complies with its policy.

The integrity protection of the handover response, as well as the handover-command guarantees that the cipher-suite negotiation is protected against bidding down (R-6).

MD, DEST, and HN get to enforce their policies such that R-5 is met. Note that it is crucial that MD sends its choice back to HN and HN in turn sends \( cs_1 \) to DEST, as otherwise HN’s policy cannot be enforced.

5.1.2 Network-Initiated \( k \)-th-order HEPB Handover

In this section, we present the history-enriched, policy-based handover approach for network-initiated \( k \)-th-order \((k \geq 1)\) handover procedures with HN or FN as anchor.\(^{10}\) The proposed procedure differs for the three handover control types only in the key-derivation method, the content of the security context, and the protection of the handover-command message sent from HCN to MD. We therefore discuss all of the handover control types at once and differentiate between them only if necessary. We reuse the handover procedure modeled in Figure 3.13 to describe the history-enriched, policy-based procedure.

5.1.2.1 Context History

In order to enable DEST to base its decision on whether to accept or refuse a handover request on the initial roaming security suite \((r)ss_0\), the previously used cipher suites \(cs_1, \ldots, cs_{k-1}\) and the key-derivation function \(kd_0\) (HN and AN-controlled cases) or \(kd_0, \ldots, kd_{k-1}\) (SRC-controlled case) used to derive \(K_k\) from previously used master session keys (R*-1), we include

\[
\text{history}_{k-1} = ((r)ss_0, kd_0, [kd_1, \ldots, kd_{k-1}], cs_1, \ldots, cs_{k-1}, T_{k-1})
\]

\(= ssh_{k-1}\)

in the security context \(S_k\) transferred from HCN to DEST during a \(k\)-th-order handover. HCN also adds \(T_{k-1}\) to the context to provide information on the total lifetime of the initial security context at the time of initiation of the \(k\)-th-order handover (R*-4).

5.1.2.2 Security Policies

We assume that there is a maximum number \(h \in \mathbb{N}\) of subsequent handover specified for each technology. The value \(h\) determines how long context histories can maximally be.

For each \(1 \leq k \leq h\), that is up to the maximum number of subsequent handover, HCN, MD and any destination network DEST have policies with respect to the cipher suites they allow to be used after a \(k\)-th-order handover given a particular security-suite history

---

\(^{10}\)This includes first-order handover with FN as anchor.
They express these policies by pre-defining sets $CS_{HCN|ssh_{k-1}}$, $CS_{MD|ssh_{k-1}}$, and $CS_{DEST|ssh_{k-1}}$ of cipher suites $cs = (ke, em, im)$ for any possible security suite history $ssh_{k-1}$. HCN, MD, and DEST pre-define these sets to be empty if and only if they do not allow for handover for a particular security suite history $ssh_{k-1}$ at all. It is important to note, that HCN, MD and DEST either allow or disallow handover for a certain security-suite history $ssh_{k-1}$. If they allow handover after a history $ssh_{k-1}$ they may have preference with respect to the cipher suite $cs_k$ that will be used after a $k$-th-order handover. However, the evaluation of $ssh_{k-1}$ is predetermined and fixed by the each handover participant independently.

Furthermore, MD, HCN, and DEST each have a policy, setting an upper boundary on how long the same initial security context may be used. To express this policy, MD, HCN, and DEST each define a threshold $Tr_{MD}$, $Tr_{HCN}$, and $Tr_{DEST}$. While $h$ is only a counter of the number of subsequent handover, the thresholds $Tr_X$ measure the lifetime of an initial security context in time units, as well as the amount of data that was so far encrypted and integrity-protected with keys derived from the initial master key $K_0$.

### 5.1.2.3 Handover Agreement

HCN enters handover agreements with destination networks DEST. As in the first-order handover case, these handover agreements regulate the terms and conditions for HCN-controlled $k$-th-order handover ($1 \leq k \leq h$).

The handover agreement includes an exchange of credentials that allow HCN and DEST to establish an authenticated and encrypted channel.

As part of the handover agreement, DEST commits to its handover policy by means of supersets $CS^*_{DEST|ssh_{k-1}} \supset CS_{DEST|ssh_{k-1}}$ for each security suite history $ssh_{k-1}$ and any $1 \leq k \leq h$. By means of this commitment, DEST commits that it will refuse a $k$-th-order handover request including $ssh_{k-1}$ unless a cipher suite $cs_k \in CS^*_{DEST|ssh_{k-1}}$ can be negotiated to be used after handover. Committing to a superset rather than $CS_{DEST|ssh_{k-1}}$ provides DEST with some degree of flexibility. It allows DEST to further restrict its policy over time, but also to explicitly refuse handover for particular security suite histories already at the time of entering the handover agreement. In turn, the commitment allows HCN to pre-select candidate destination networks according to their commitments. This allows HCN to avoid requesting handover in vain. As in the first-order case, DEST commits to $CS$ if it does not want to make any commitment at all and commits to $CS^*_{DEST|ssh_{k-1}} = \emptyset$ if it does not allow for $k$-th-order handover given a particular security-suite history $ssh_{k-1}$.

Similarly, DEST commits to an upper boundary $u_{DEST}$ on its threshold $Tr_{DEST}$. By this commitment, DEST commits to refusing handover requests that include a lifetime indicator $T_k \geq u_{DEST}$. This commitment again helps HCN to pre-select candidate destination networks and allows DEST to lower its threshold without prior notice to HCN.
5.1.2.4 Trust Assumption

Throughout the rest of this chapter we assume $\text{DEST}_k$ trusts HCN on all information received by HCN. In particular we assume that in the HN-controlled case, $\text{DEST}_k$ trusts that HN included the correct information on the security context history in the security context $S_k$. This assumptions seems very lax, as $\text{DEST}_k$ and HN establish a trust relationship when entering a handover agreement and have to trust each other to some extend anyway, e.g., as HN generates the key $K_{k,1}$. $\text{DEST}_k$ subsequently uses. Similarly, we assume that in the AN-controlled case, $\text{DEST}_k$ trusts that AN included the correct information on the security context history in the security context $S_k$. Moreover we assume that in the SRC-controlled case $\text{DEST}_k$ trusts that SRC included the correct information into the security context $S_k$. Note that in the SRC-controlled case this assumption again leads to transitive trust relationships between the subsequently serving network, which corresponds to the fact that $\text{DEST}_k$ only has a handover agreement with SRC in this case.

5.1.2.5 Key Derivation and Security Context

The key derivation depends on who controls the handover.

**SRC-Controlled Handover.** In an SRC-controlled handover, MD, $\text{SRC}_{k-1}$, and $\text{SRC}_k$ negotiate a key-derivation function $kd_{k-1}$ as part of the security-mechanism negotiation on the $(k-1)$-st order handover. $kd_{k-1}$ is included in the security context $S_k$, which $\text{DEST}_k$ receives from $\text{SRC}_k$ during the $k$-th order handover. We require $kd_{k-1}$ to be a pre-image resistant hash function (see [120] for definition) that takes $K_{k-1}$, DEST$_k$’s identity and a fresh random number $\text{RAND}$ as input and outputs a derived master session key $K_k$.

The pre-image resistance guarantees that knowledge of $K_k$ does not reveal any information on any previously used master session key $K_j (0 \leq j \leq k-1)$ ($R^*-2$). However, knowledge of any previous key and the identities of the previously serving networks implies knowledge of $K_k$. Consequently, $R^*-3$ cannot be met by this key-derivation method. In particular, $\text{SRC}_k$ and any previously serving network $\text{SRC}_j (1 \leq j \leq k)$ gain knowledge of $K_k$.

The $k$-th order security context on an SRC-controlled handover is

$$S_k = (K_k, \text{history}_{k-1}, Tr_{SRC}),$$

where

$$\text{history}_{k-1} = ((r)ss_0, kd_0, kd_1, \ldots, kd_{k-1}, cs_1, \ldots, cs_{k-1}, T_{k-1}).$$

**AN-Controlled Handover.** In an AN-controlled handover, MD and AN negotiate a pre-image resistant hash function as key-derivation function $kd_0$ during the negotiation of the initial security suite $ss_0$. This key-derivation function is included in any subsequent security-context transfer $S_k (1 \leq k \leq h)$. Again, $kd_0$ takes $K_0$, $\text{DEST}_k$’s identity, and
a fresh random number $RAND$ as input and outputs a master session key $K_k$. The pre-image resistance guarantees that R-2 is met. The fact that AN (and MD) derive $K_k$ from $K_0$ additionally guarantees that knowledge of $K_j$ ($1 \leq j \leq k-1$) does not reveal any information on $K_k$. However, it is important to note that knowledge of $K_0$ reveals all subsequently used $K_j$ ($1 \leq j \leq k$). Consequently, this key-derivation method meets R-3 in part.

The $k$-th-order security context on an AN-controlled handover is

$$S_k = (K_k, history_{k-1}, Tr_{AN}),$$

where

$$history_{k-1} = ((r)ss_0, kd_0, cs_1, \ldots, cs_{k-1}, T_{k-1}).$$

**HN-Controlled Handover.** In an HN-controlled handover, MD and HN negotiate a key-derivation function $kd_0$ as part of the pre-registration process. HN either obtains knowledge of the initial master session key $K_0$ during the authentication between AN and MD, or AN transfers $K_0$ to HN as part of the handover-indication message. $K_k$ is derived from $K_0$ in the same way as in the AN-controlled case. Again, this key-derivation method meets R-2 and guarantees that the knowledge of $K_j$ ($1 \leq j \leq k-1$) does not reveal any information on $K_k$. However, AN and HN may gain knowledge of all of the subsequently used master session keys. Consequently, R-3 is again met in part.

The $k$-th-order security context on an HN-controlled handover is

$$S_k = (K_k, history_{k-1}, Tr_{HN}),$$

where $history_{k-1}$ is defined as in the AN-controlled case.

### 5.1.2.6 Network-Initiated $k$-th-order Handover Procedure

For an overview on the subsequent handover procedure, we refer to Figure 3.13. As in the first-order case, we will discuss various methods to negotiate the cipher suite $cs_k$ to be used after handover at the end of this section. In the following description of the handover procedure itself, we only describe the common interface of the negotiation methods.

Before a $k$-th-order handover, MD is connected to SRC$_k$. MD and SRC$_k$ use the encryption mechanism $em_{k-1}$ and the integrity-protection mechanism $im_{k-1}$ negotiated during the $(k-1)$-st-order handover from SRC$_{k-1}$ to DEST$_{k-1} = SRC_k$ in order to protect the data and control traffic between them.

In the security-mechanism negotiation phase (1), MD and SRC$_{k-1}$ negotiate on the cipher suite to use after handover. In most of the negotiation methods we present, MD simply sends $CS_{MD}|ssh_{k-1}$ to SRC$_k$ in phase (1).

Upon detecting a handover reason, SRC$_k$ sends a handover-indication message to HCN. This message includes measurement data provided by MD and SRC$_k$ on the reception level
of surrounding NAPs. Depending on the negotiation method, the handover indication may additionally include $CS_{MD|ssh_0}$.

HCN determines an ordered list $L = \{\text{DEST}_1^k, \ldots, \text{DEST}_n^k\}$ of candidate destination networks with which it has a handover agreement. HCN picks the first candidate network $\text{DEST}_1^k$ in its list and checks whether $u_{\text{DEST}_1^k} > T_{k-1}$. HCN then checks whether $CS_{\text{DEST}_1^k|ssh_{k-1}} \neq \emptyset$. If either of these checks fail, HCN restarts the selection process with $\text{DEST}_2^k$. If both checks are successful, HCN sends a handover request to $\text{DEST}_1^k$.

The handover-request message sent from HCN to any candidate network $\text{DEST}_i^k$ (1 $\leq$ $i$ $\leq$ $n$) includes the security context $S_k$, as well as the identities of MD, $\text{DEST}_i^k$, and HCN. The handover-request message is authenticated and encrypted by means of keys agreed upon between HCN and $\text{DEST}_i^k$ based on the credentials established on entering the handover agreement (R*-7).

Upon receipt, $\text{DEST}_i^k$ first checks whether it received the handover request from an HCN it indeed has a handover agreement with. $\text{DEST}_i^k$ then checks whether $T_{k-1} < T_r_{\text{DEST}_i^k}$ and whether $CS_{\text{DEST}_i^k|ssh_0} \neq \emptyset$. If either of these checks fail, $\text{DEST}_i^k$ sends a negative handover response back to HCN. If all of these checks are successful, $\text{DEST}_i^k$ sends a positive handover response back to HCN.

Upon receipt of a positive handover response from $\text{DEST}_i^k$, HCN starts to negotiate a cipher suite $cs_k$ with $\text{DEST}_i^k$ using one of the negotiation methods described below. If the negotiation is not successful, HCN restarts the selection process with the next candidate network in $L$. If the negotiation is successful, HCN selects $\text{DEST}_i^k$ as destination network $\text{DEST}_k$ for the $k$-th-order handover and sends a handover command message to MD including $\text{DEST}_k$’s identity and the fresh random number $RAND$ used for key-derivation. Depending on the negotiation method used, the handover command message additionally includes information on the cipher suite(s) negotiated between HCN and $\text{DEST}_i^k$ in the negotiation phase (2). The integrity protection of the handover command message (R*-8) depends on who controls the handover:

**SRC-Controlled Case.** In the SRC-controlled case, the handover command message is integrity-protected by means of the integrity-protection mechanism $im_{k-1}$ and the integrity-protection key $IK_{k-1}$.

**AN-Controlled and HN-Controlled Cases.** In the AN-controlled and the HN-controlled cases, the handover-command message is integrity-protected by means of the initial integrity-protection mechanism $im_0$ and the initial integrity-protection key $IK_0$.

Upon receipt of the handover command, MD checks whether $T_{k-1} \leq T_{r_{MD}}$. In the positive case, MD extracts $RAND$ and derives $K_1$ from $K_0$, $RAND$ and $\text{DEST}_k$’s identity. Then, MD and $\text{DEST}_k$ associate and use the negotiated $ke_k$ to establish data-protection keys $EK_k$ and $IK_k$ and subsequently use the negotiated $em_k$ and $im_k$ for encryption and integrity protection.
Security-Mechanism Negotiation. As in the first-order case, HCN, MD, and DEST\_k can negotiate the cipher suite to use after handover in various different ways. In this section, we generalize the five negotiation methods introduced in Section 5.1.1.5 to the subsequent handover case.

**Method 1.**

*HN-Controlled Case.* HN stores the sets of cipher suites $CS\_MD\_{ssh_{k-1}}$ ($0 \leq k \leq h$) of all its pre-registered MDs.

*AN-Controlled Case.* HN stores the sets of cipher suites $CS\_MD\_{ssh_{k-1}}$ ($0 \leq k \leq h$) of all its pre-registered MDs. In case the anchor network is FN, HN transfers all of these policies to AN right after or as part of the authentication between FN and MD.

*SRC-Controlled Case.* SRC\_k obtains MDs policy expressions $CS\_MD\_{ssh_{i-1}}$ ($k \leq i \leq h+1$) from SRC\_k-1 during the ($k-1$)-st order handover from SRC\_k-1 to SRC\_k ($2 \leq k \leq h$). AN obtains MDs policies either during the pre-registration (AN = HN) or during the initial authentication between AN and MD (AN = FN).

In all three handover control types, the negotiation proceeds as follows: before sending a handover request to DEST\_i\_k on a $k$-th-order handover, HCN computes the intersection

$$Nego_1 = CS\_HCN\_{ssh_{k-1}} \cap CS\_MD\_{ssh_{k-1}} \tag{5.1}$$

HCN then includes $Nego_1$ in the handover request sent to DEST\_i\_k. DEST\_i\_k computes the intersection $Nego_2 = Nego_1 \cap CS\_DEST\_i\_k\_{ssh_{k-1}}$ and selects one of the elements of $Nego_2$ to be the cipher suite $cs_k$. DEST\_i\_k returns $cs_k$ back to HCN as part of the handover response. HCN checks whether $cs_k$ complies with its policy. If this check is not successful, HCN restarts the destination network selection process with DEST\_i\_k+1. Otherwise, HCN selects DEST\_i\_k to be the next destination network DEST\_k and includes $cs_k$ in the handover command to MD. Upon receipt of the handover command, MD checks whether $cs_k$ complies with its policies. If this is not the case, MD drops the connection. If the compliance check is successful, MD tries to associate with DEST\_k.

As DEST\_i\_k selects the cipher suite to use and HN and MD both check that DEST\_i\_k’s choice complies with their policies, MD, HCN, and DEST\_i\_k can enforce that the negotiated cipher suite $cs_k$ complies with their policies. Consequently, R\_5 is met by this negotiation method.

In the AN-controlled and the SRC-controlled cases, the transfer of $CS\_MD\_{ssh_{k-1}}$ from HN to AN respectively from SRC\_k-1 to SRC\_k is integrity-protected by means of keys agreed upon based on the credentials exchanged on entering the handover agreement. Additionally, the handover-response message is integrity-protected by means of keys agreed upon between DEST\_i\_k and HCN based on the credentials exchanged on entering the handover agreement. The handover-command message sent from HCN and MD is integrity-protected as already discussed. Consequently, the negotiation Method 1 is protected against bidding-down attacks (R\_6).

**Method 2.** Instead of having HN pre-store all policies of its pre-registered users, MD sends the set of cipher suites $CS\_MD\_{ssh_{k-1}}$ it allows to be used after a $k$-th-order handover given
ssh_{k-1} to SRC_k in Phase (1).

SRC_k forwards MD’s policy to HCN (if different from SRC) as part of the handover-indication message sent to HCN after detecting a handover reason. The rest of Method 2 is the same as in Method 1.

Method 2 meets R*-5 in the same way as Method 1. To meet R*-6, MD has to protect the integrity of $CS_{MD}|_{ssh_{k-1}}$ during its transfer to HCN via SRC_k. In the HN-controlled and the AN-controlled cases, MD uses the integrity-protection mechanism $im_0$ and the integrity key $IK_0$ to protect the transfer. In the SRC-controlled case, MD protects the transfer by means of $im_{k-1}$ and $IK_{k-1}$. In combination with the integrity protection of handover response and handover command as described in Method 1, R*-6 is met.

**Method 3.** MD again sends $CS_{MD}|_{ssh_{k-1}}$ to SRC_k in Phase(1). After detecting a handover reason, SRC_k forwards MD’s policy to HCN. DEST_i_k sends its policy $CS_{DEST_i}|_{ssh_{k-1}}$ to HCN in Phase (2), included in the handover response. HCN computes the intersection $Nego_1$ of $CS_{MD}|_{ssh_{k-1}}$, $CS_{HCN}|_{ssh_{k-1}}$, and $CS_{DEST_i}|_{ssh_{k-1}}$ and orders it according to its preference order. HCN then orders the result according to the preference order of MD and finally according to the preference order of DEST_i_k. HCN then randomly selects $cs_k$ back to DEST_i_k in an integrity-protected message and includes $cs_k$ in the integrity-protected handover command it sends to MD. Upon receipt of $cs_k$, MD and DEST_i_k check whether HCN’s choice indeed complies with their policies. Consequently, this method meets R*-5.

MD protects the transfer of $CS_{MD}|_{ssh_{k-1}}$ to HCN as described in Method 2. This, together with the integrity protection of the messages indicating HCN’s choice of $cs_k$, protects the negotiation of $cs_k$ against bidding-down attacks (R*-6).

**Method 4.** MD again sends $CS_{MD}|_{ssh_{k-1}}$ to SRC_k in Phase(1) and SRC_k forwards it to HCN in the handover request. HCN computes the intersection

$$Nego_1 = CS_{MD}|_{ssh_{k-1}} \cap CS_{HCN}|_{ssh_{k-1}}$$

and includes it in the handover request sent to DEST_i_k. DEST_i_k computes the intersection of $Nego_1$ with its own $CS_{DEST_i}|_{ssh_{k-1}}$ and orders the result according to its own preference order. DEST_i_k then orders the cipher suites it prefers most according to HCN’s preference order. Finally, if HCN prefers more than one cipher suite most, DEST_i_k orders these according to MD’s preference order:

$$cs_k \in_R \max_{\leq_{DEST_i_k}} \left\{ \max_{\leq_{MD}} \left\{ \max_{\leq_{HCN}} \{ Nego_1 \cap CS_{DEST_i}|_{ssh_{k-1}} \} \right\} \right\}.$$
Method 5. DEST\(_k^i\) sends \(CS_{\text{DEST}^i_k}\)\(_{ssh_{k-1}}\) to HCN as part of the handover response and protects this message by means of keys agreed upon based on the credentials exchanged between HCN and DEST\(_k^i\) on entering the handover agreement. HCN computes the intersection of its own \(CS_{HCN}\)\(_{ssh_{k-1}}\) with the received set and sends the result to MD. MD finally computes the intersection

\[\text{Nego}_3 = CS_{\text{DEST}^i_k}\)\(_{ssh_{k-1}}\cap CS_{MD}\)\(_{ssh_{k-1}}\cap CS_{HCN}\)\(_{ssh_{k-1}}\]

and orders this set first according to its own preferences. If it regards more than one cipher suite equally well, MD orders these maximal suites according to HCN’s preference order and finally, if HCN prefers more than one cipher suite most, according to DEST\(_k^i\)’s preference order. MD then randomly chooses \(cs_k\) from the remaining cipher suites:

\[cs_k \in_R \max_{\leq_{\text{DEST}^i_k}} \{ \max_{\leq_{HCN}} \{ \max_{\leq_{MD}} \{ \text{Nego}_3 \} \} \} \] .

MD sends \(cs_k\) back to HCN and HCN informs DEST\(_k^i\) of MD’s choice. Thus, HCN and DEST\(_k^i\) get to refuse MD’s choice of \(cs_k\) and R*-5 is again met. DEST\(_k^i\) protects the integrity of the handover-response message and HCN protects the integrity of the handover-command message as discussed in Method 3. This protects Method 5 against bidding-down attacks (R*-6).

5.1.3 Differences in the Mobile-Initiated Case

The context history, the security policies, the handover agreements, and the key derivation, for a mobile-initiated procedure are the same as for a network-initiated handover. In particular, the key-derivation methods for different handover control types fully meet R*-2 and meet R*-3 in part, as in the network-initiated case. As opposed to the network-initiated case, the security context in the mobile-initiated case includes the random number \(RAND\) generated by HCN to derive the master session key \(K_k\). \(RAND\) is integrity-protected by means of a key shared between HCN and MD (\(IK_{k-1}\) in the SRC-controlled case, \(IK_0\) in the AN-controlled and the HN-controlled cases). The same holds for R*-4, as \(T_{k-1}\) is included in the context history. However, the procedures themselves, as well as the negotiation of the cipher suite to use after handover, differs from the network-initiated case.

HCN Notified by MD. In case HCN is notified by MD (see Figure 3.15, the handover-indication message sent from MD to HCN is integrity-protected with the help of \(IK_0\) and \(im_0\) in the AN-controlled and HN-controlled cases or \(IK_{k-1}\) and \(im_{k-1}\) in the SRC-controlled case (R*-8). The handover-indication message includes the identities of MD, HCN, and DEST\(_k^i\).

Upon receipt of a handover indication message, HCN checks whether it received this message from MD correctly integrity-protected. HCN then checks whether it indeed has a handover agreement with DEST\(_k^i\). If both of these checks are successful, HCN proceeds as in the network-initiated case, i.e., HCN checks whether \(CS_{\text{DEST}^i_k}\)\(_{ssh_{k-1}}\) is not empty. If this
check is successful, HCN sends a handover request to DEST\textsubscript{i} including the same information as in the network-initiated case. The handover request is encrypted and integrity-protected as in the network-initiated case (R*-7).

Upon receipt of a handover request from HCN, DEST\textsubscript{i} decides whether to accept or refuse the request in the same way as in the network-initiated case, based on the context history included in the handover request (R*-1).

DEST\textsubscript{i} acknowledges its response in the handover response message sent to MD. In the positive case, DEST\textsubscript{i} includes the integrity-protected random number RAND in the handover response message.

Here we give only one example for a method to negotiate the cipher suite to be used after handover: MD sends $CS\textsubscript{MD}|_{ssh_{k-1}}$ to HCN via SRC\textsubscript{k} included in the integrity-protected handover-indication message. HCN computes the intersection

$$Nego_1 = CS\textsubscript{MD}|_{ssh_{k-1}} \cap CS\textsubscript{HCN}|_{ssh_{k-1}}$$

and includes Nego\textsubscript{1} in the handover-request message sent over an encrypted and authenticated channel to DEST\textsubscript{i}. DEST\textsubscript{i} computes the intersection

$$Nego_2 = Nego_1 \cap CS\textsubscript{DEST\textsubscript{i}}|_{ssh_{k-1}}.$$ 

DEST\textsubscript{i} selects one of the cipher suites in Nego\textsubscript{2} as $cs_k$ and sends its choice back to HCN in an integrity-protected message. HCN checks whether $cs_k$ complies with its policy. If this is the case, it sends a message including $cs_k$ back to DEST\textsubscript{i}. HCN protects the integrity of this message by means of $IK_0$ and $im_0$ (HN-controlled and AN-controlled cases) or $IK_{k-1}$ and $im_{k-1}$ (SRC-controlled case). DEST\textsubscript{i} includes this message and its integrity protection in the handover-response message it sends to MD after successful association. Upon receipt of the handover response, MD checks HCN’s integrity protection on the message including $cs_k$. If this is successful, MD checks whether $cs_k$ complies with its policies. If either of these checks fail, MD drops the connection to DEST\textsubscript{i}. The integrity protection of HCN’s approval of $cs_k$, the fact that MD checks the correctness of the integrity protection, and the fact that MD checks whether $cs_k$ complies with its policy guarantee that R*-5 is met. Moreover, the fact that all messages containing policy-related information are integrity-protected guarantees that this negotiation method is protected against bidding down attacks (R*-6).

**HCN Notified by DEST\textsubscript{i}**. In case HCN is notified by DEST\textsubscript{k} (see Figure 3.14), MD sends the handover indication message to DEST\textsubscript{k} right after association, and DEST\textsubscript{k} forwards it to HCN. The rest of the procedure is the same as if HCN is notified by MD.

One way to negotiate the cipher suite $cs_k$ to be used after handover is the following: MD sends $CS\textsubscript{MD}|_{ss}_{ssh_{k-1}}$ to DEST\textsubscript{i} after the association. MD protects the integrity of this message by means of $IK_0$ and $im_0$ (HN-controlled and AN-controlled cases) or $IK_{k-1}$ and $im_{k-1}$ (SRC-controlled case). DEST\textsubscript{i} sends $CS\textsubscript{MD}|_{ssh_{k-1}}$ (including its integrity protection by HCN) and $CS\textsubscript{DEST\textsubscript{i}}|_{ssh_{k-1}}$ to HCN in the integrity-protected handover-indication message...
it sends to HCN. Upon receipt of this message, HCN checks the correctness of the integrity protection of DEST\(_k\) and MD. If both checks are successful, HCN computes the intersection of the three sets \(CS_{HCN}|_{ssh_{k-1}}, CS_{MD}|_{ssh_{k-1}},\) and \(CS_{DEST}\_{k}|_{ssh_{k-1}}\) and chooses one of the cipher suites in the intersection as \(cs_k\). HCN includes its choice in the integrity-protected handover-request message to DEST\(_k\). HCN also adds integrity protection to \(cs_k\), computed by means of \(IK_0\) and \(im_0\) (HN-controlled, AN-controlled case) or \(IK_{k-1}\) and \(im_{k-1}\) (SRC-controlled case) to the handover request. Upon receipt of the handover request, DEST\(_k\) checks whether \(cs_k\) complies with its policy. If this is the case, DEST\(_k\) adds \(cs_k\) and the integrity protection of \(cs_k\) by HCN to the handover-response message sent to MD. Upon receipt of the handover response, MD checks whether \(cs_k\) complies with its policy.

As HCN selects \(cs_k\) and DEST\(_k\) and MD check the compliance of \(cs_k\) with their respective policies, R*-5 is met by this negotiation method. Moreover, the integrity protection on all policy-related messages guarantees that the negotiation of the cipher suite \(cs_k\) is protected against bidding-down attacks, thus meeting R*-6.

### 5.1.4 Handling Long Histories and Large Amounts of Policy Data

As described so far, the key history \(ssh_k\) transferred on a \(k\)-th-order handover includes the initial security suite \(ss_0\), all subsequently used cipher suites \(cs_1, . . . , cs_{k-1}\) in the order they were used, and the key-derivation function \(kd_0\) (HN-controlled and AN-controlled cases) or the key-derivation functions \(kd_0, . . . , kd_{k-1}\) (SRC-controlled case). With every subsequent handover, the context history is thus extended by one cipher suite \(cs\) and in the SRC-controlled case also by one key-derivation function \(kd\). The actual number of occurring subsequent handover depends on the size of each subsequently serving network, the duration of an ongoing connection, the velocity of MD, and the path MD takes through each of the networks. The smaller the coverage area of the subsequently serving networks is and the higher the velocity of a MD that travels straight through each network, the more subsequent handover can be expected.

For example, let us consider subsequent handover procedures between different wide-area network providers that each cover an area of 15 km radius and a MD that travels at 100 km/h directly through the providers’ networks. Within the 90 minutes required to watch a streaming movie, MD maximally crosses the networks of five providers. Thus, the key history on the fourth-order handover includes the initial security suite \(ss_0\), as well as \(cs_1, cs_2\), and \(cs_3\). Additional subsequent handover do not take place as the only ongoing service use, the video streaming, completes within the 90 minutes.

The above example indicates that in some cases, transferring the complete history is feasible. However, for other scenarios, transferring the complete histories can result in a too-large amount of data to transfer and process upon handover. Imagine, for example, a densely populated metropolitan area in which thousands of private local area network providers (e.g., private persons) operate a network with a coverage area as small as 100 m\(^2\). A MD traveling with 50 km/h through these networks will travel through 750 networks within the 90 minutes of a streaming movie. In this scenario, it may not be feasible to transfer and process the complete history on each subsequent handover.
With the length of the security-suite history and the maximum number of subsequent handover \( h \), the number of sets of cipher suites that have to be pre-defined and stored by MD and each network grows. For each \( 1 \leq k \leq h \), the number of possible cipher suites in \( ssh_{k-1} \) is
\[
|CS|^k
\]
such that each participant has to set and store
\[
\sum_{k=1}^{h} |CS|^k \cdot |(R)A \times (R)KA|
\]
sets of cipher suites it allows to be used after handover. The number of cipher-suite sets thus grows exponentially with the number of subsequent handover.

The above scenario, as well as the amount of policy data that has to be stored and processed during handover if each cipher suite is listed in the security-suite history in the order it was used, motivate us to think of ways to compress the information included in a key history. Whether or not a compression is necessary and which compression method is most suitable has to be decided for each technology and application scenario separately. Obviously, any compression method reduces the granularity in which the participants of subsequent handover can express their policies. Consequently, whenever feasible, no compression should be used.

The easiest way to reduce the length of a key history is to omit subsequent uses of the same cipher suite. This compression method is lossless, as the use of the same cipher suite between MD and one network or between MD and two different networks does not carry any additional information. This compression method can therefore be used without further concern. However, this compression method may still result in long histories, e.g., in case MD is handed back and forth between two networks resulting in the alternate use of two different cipher suites and does not reduce the amount of policy data that has to be pre-defined and stored.

A second method to reduce the length of a key history is to omit the order and frequency of appearance of the subsequently used cipher suites. A reappearing cipher suite is thus not mentioned again in the transferred history. This compression method guarantees small key histories whenever the overall number of available cipher suites \(|CS|\) for the given technology is small.\(^{11}\) Moreover, this method considerably reduces the length of histories in case MD moves on the border line between two networks and is subsequently handed back and forth between them.

Omitting the order and frequency of the appearance of a cipher suite additionally decreases the number of sets of cipher suites that have to be pre-defined considerably. On a

\(^{11}\)As opposed to the authentication protocols, the cipher suites used in a wireless technology have to be standardized, as otherwise handover and roaming cannot take place. As a consequence, wireless technologies typically use fewer than 10 cipher suites such that this compression method seems feasible for state-of-the-art technologies.
5.1 HEPB SCT with Key Derivation

$k$-th-order handover MD, HCN, and DEST$_k$ can encounter

$$\sum_{i=1}^{|CS|} \binom{|CS|}{i} - \sum_{i=k+1}^{|CS|} \binom{|CS|}{i}$$

cipher suites such that MD, and each network overall maximally has to pre-define and store

$$2^{|CS|} \cdot |(R)A \times (R)KA|$$

sets of cipher suites.\footnote{If $h \leq |CS|$ the number of sets of cipher suites that have to be pre-defined is $\sum_{i=1}^{|CS|} \binom{|CS|}{i} - \sum_{i=h+1}^{|CS|} \binom{|CS|}{i}$.}

By omitting the order of appearance of a cipher suite in the history, the participants lose the ability to base their policies on certain patterns of subsequently used cipher suites. For example, they can no longer forbid handover for a certain pattern of subsequently used cipher suites only. Instead, all handover with histories that include all of the different cipher suites in the pattern have to be forbidden. If, for example, a network provider wants to forbid handover to its network as destination network if the history includes the pattern $(cs_1^1, cs_2^2, cs_3^1)$ that is if $cs^2$ has been used in between two usages of $cs^1$, then the destination network now has to forbid all handover with a history that includes $cs^1$ and $cs^2$.\footnote{Note that we assume here that the participants of a handover pre-define their policies and either allow or disallow handover with a certain security context history.}

Whether or not this is a serious loss for the participants depends on whether there are attacks against certain patterns rather then certain combinations of cipher suites in connection with a particular key-derivation function $kd_0$. Currently, the security of protection mechanisms like encryption mechanisms or integrity-protection mechanisms are typically studied independently from other mechanisms. For example, the security of an encryption mechanism is typically studied under the assumption that the encryption key is randomly and uniformly chosen from the space of encryption keys. Assumptions on the strength of mechanisms are thus typically formed independently of other mechanisms. However, the use of a security-context transfer with key derivation makes it necessary to study the impact of combinations and sequences of security mechanisms used with related session keys. Cryptanalysis of combinations and sequences of mechanisms are not well-studied in the literature so far. Although we acknowledge here the relevance of such studies, they have to be done for each technology’s set of security suites and cannot be addressed with the level of abstraction in this part of the thesis. Based on state-of-the-art cryptoanalyzing tools, omitting the frequency and order of occurrence of the cipher suites in the security suite histories is the method of choice to reduce the amount of policy data each handover participant is required to store.

Another obvious possibility to restrict the length of the transferred histories is to set the maximal number $h$ of subsequent handover in a way that all histories of length $h$ can easily be handled. The second scenario above indicates that this alone may not solve the problem adequately for all cases, as it may restrict seamless handover to short time periods.
If the security mechanisms for a technology are well studied, it may be possible to assign a security level to each possible cipher suite in a way that is accepted by all providers and users. Providers and users then express their respective policies with the help of these level assignments. Instead of including the complete history in the security context, HN includes the level of the lowest level cipher suite thus far used in the security context. On a $k$-th-order handover, DEST$_k$ then decides whether to accept or reject the handover request based on the lowest security level thus far used. With this compression method, the participants of a handover not only lose the ability to protect against certain weak combinations of security mechanisms, but this method also makes it difficult for DEST$_k$ to react to the unexpected break of, e.g., a certain encryption mechanism. More granularity can be provided by assigning security levels to each encryption mechanism, integrity-protection mechanism, and key-establishment process separately. HN then includes $a_0$, $ka_0$, $kd_0$, as well as the level of the lowest-level encryption mechanism, the level of the lowest-level integrity-protection mechanism, and the level of the lowest-level key establishment used thus far into the security context. DEST$_k$ can then forbid handover based on each of the levels rather than on the overall level.

However, in current wireless access technologies the number of specified cipher suites $|CS|$ is very restricted. For example, in GSM $|CS| = 4$, in UMTS $|CS| = 2$ and in 802.11i-protected WLANs $|CS| = 3$. Consequently, omitting the order and frequency of cipher suites in the history seems feasible and is by far the better choice, as it leaves the full power over the evaluation of a security-suite history to each participant.

### 5.2 HEPB SCT with Key Agreement

The history-enriched, policy-based security-context transfer with key agreement differs from the one with key derivation only in the content of the context history and the way the master key is agreed upon.

In case a fresh master key is agreed upon between SRC$_k$ and MD before a $k$-th-order handover by means of a roaming key-agreement protocol $(r)ka_k$ and the last authentication protocol used between MD and any previously serving networks was $(r)a^*$, the context history $history_{k-1}$ is

$$history_{k-1} := ((r)a^*, (r)ka_k, em_{k-1}, im_{k-1}, ke_{k-1})(=: ssh_{k-1})$$

in the SRC-controlled case and

$$history_{k-1} := ((r)a^*, (r)ka_k, em_0, im_0, ke_0)(=: ssh_{k-1})$$

in the HN-controlled as well as in the AN-controlled case. For all three control types, the security context is

$$S_k := (K_k, history_{k-1}, Tr_{HCN}).$$

In case multiple initial keys are generated during the initial roaming key-agreement protocol $(r)ka_0$ the context history is

$$history_{k-1} := ((r)a_0, (r)ka_0, em_{k-1}, im_{k-1}, ke_{k-1})(=: ssh_{k-1})$$
in the SRC-controlled case and
\[\text{history}_{k-1} := ((r)a_0, (r)ka_0, em_0, im_0, ke_0)(= ssh_{k-1})\]
in the HN-controlled and the AN-controlled case.

In the HN-controlled and AN-controlled cases, the security context \(S_k\) is defined as above. However, in the SRC-controlled case, the security context \(S_k\) transferred on a \(k\)-th-order handover is
\[S_k := (K_k, K_{k+1}, K_h, Tr_{HCN}).\]

The requirements R’-1, R’-5 R’*-6, R’*-8, and R’-7 can be met in the same way as in SCT with key derivation. Requirement R’-2 that is specific to this type of SCT is met in part by the two key-agreement methods thus far suggested. In case the new key agreement is used between MD and SRC\(_k\), SRC\(_k\) and HCN get to know the master session key \(K_k\). In case multiple initial keys are generated during the initial key agreement, the situation differs for the handover control types. In the case of HN-controlled and AN-controlled handover, HN or AN gets to know all subsequently used master session keys. In the case of SRC-controlled handover, each subsequent source network gets to know all master session keys used after future handover.

In order to fully meet R’-2, we suggest a third key-agreement method in the next section.

5.3 Using Secret Sharing for Key Agreement Between \(\text{DEST}_k\) and MD During Handover

In this section, we present a key-agreement method for SCT on an HN-controlled handover that meets the requirement R’-2. It allows MD and \(\text{DEST}_k\) to agree upon a master session key to be used after handover in a way that no other party, except for MD and \(\text{DEST}_k\), gains any information on this master session key.

We assume that each HN is issued a handover certificate on an encryption key for a public-key encryption scheme (e.g., RSA or any other public-key encryption scheme that allows for a secure two-party decryption) by a trusted third party. Assuming HN has a handover agreement with \(l\) destination networks \(\text{DEST}_1, \ldots, \text{DEST}_l\), the trusted third party splits the secret handover key \(R\) corresponding to the public key in the handover certificate into \(l\) different pairs of shares \((R_{HN_i}, R_{\text{DEST}_i}) 1 \leq i \leq l\) by means of a \((2,2)\) secret-sharing scheme. The trusted third party distributes \(R_{HN_i} 1 \leq i \leq l\) to HN and each \(R_{\text{DEST}_i}\) to \(\text{DEST}_i\).

MD pre-stores the handover certificate of its HN.

Before a \(k\)-th-order handover, MD encrypts a fresh random number \(r\) with the public encryption key included in HN’s handover certificate and sends the encrypted random number to SRC\(_k\). SRC\(_k\) forwards this random number to HN in the handover-request message (network-initiated case) or the handover-indication message (mobile-initiated case). HN and \(\text{DEST}_k\) use a two-party version of the corresponding public-key decryption such that \(\text{DEST}_k\) with the help of HN can recover \(r\), but HN obtains no information on \(r\) itself.
We exemplify this approach for the RSA case. Assume HN’s public handover key is an RSA key \((n, e)\) where \(n\) is the modulus and \(e\) is the encryption key. Furthermore, let \(d\) be the secret key corresponding to \(e\) such that \(ed = 1 \mod \varphi(n)\), where \(\varphi(n) = (p - 1)(q - 1)\). Then the trusted third party splits the secret handover key \(d\) into \((d_{HN}, d_{DEST})\) for \(i = 1 \ldots l\) in the following way: the trusted third party chooses \(\omega_1, \ldots, \omega_l \in \mathbb{Z}_{\varphi(n)}\) such that \(\omega_i \neq \omega_j\) for \(i \neq j\). Then,

\[
d_{HN_i} = d + 2\omega_i \mod \varphi(n), \quad i = 1, \ldots, l
\]

\[
d_{DEST_i} = d + \omega_i \mod \varphi(n), \quad i = 1, \ldots, l.
\]

Consequently, \(d = -d_{HN} + 2d_{DEST}\) \mod \varphi(n), for all \(i = 1, \ldots, l\). The trusted third party distributes \(d_{DEST_i}\) to DEST\(^i\) and all \(d_{HN_i}\) \(i = 1, \ldots, l\) to HN. The trusted third party also keeps copies of all \(\omega_i\), as well as \(d\).

Before a \(k\)-th-order handover, MD generates a fresh random number \(r\), encrypts it with the public handover key \(e\) of its HN, and sends it to SRC\(_k\) together with an integrity protection generated by means of \(IK_0\) and \(im_0\). Upon detecting a handover reason, SRC\(_k\) includes \(r_e\) in the handover request (or the handover indication) to HN. HN checks the validity of the integrity protection on \(r_e\) by MD. In case this check is successful, HN uses its split \(d_{HN_i}\) of the secret handover key corresponding to DEST\(^i\)\(_k\) to compute \(r_{ed_{HN}}\) and includes this value instead of \(K_k\) in the handover request sent to DEST\(^i\)\(_k\). Upon receipt of \(r_{ed_{HN}}\), DEST\(^i\)\(_k\) uses its part of the secret handover key \(d_{DEST_i}\) in order to decrypt \(r\):

\[
r = r_e(-d_{HN_i})^e(2d_{DEST_i})^e.
\]

DEST\(^i\)\(_k\) and MD then compute \(K_k\) by means of a pre-image resistant hash function \(kd_k\) with \(r\) and the identities of MD and DEST\(^i\)\(_k\) as input. As only DEST\(^i\)\(_k\) (and the trusted third party) has knowledge of the secret key share \(d_{DEST_i}\), an attacker intercepting \(r_e\) and \(r_d\) cannot gain any information on \(r\). In particular, neither SRC\(_k\) nor HN, and not even those two parties in collaboration, can recover \(r\). Consequently, this key-agreement method meets R'-2. It is interesting to note that the security-context transfer used by this key-derivation method does not need to be encrypted, as it does not include any secret information (R'-7).

Instead of a trusted third party, the key splitting may be performed in HN. However, in this case, although the master session key \(K_k\) is not computed by HN, HN has knowledge of the full secret handover key \(d\) and may thus compute \(K_k\). Consequently, in this case, R'-2 cannot fully be met. However, this key-agreement variant still has the advantage that HN is not required to generate \(K_k\) and no encryption of the security-context transfer is required. This makes DEST\(^i\) less dependent on the way HN protects the storage of secret keys.

In Chapter 12, we present how the key-agreement method described here can be used to agree upon a master session key \(K_k\) between MD and DEST\(_k\) on an HN-controlled \(k\)-th-order handover between two WLAN networks and show how this key-agreement approach can be integrated with the 802.11i security architecture. To the best of the author’s knowledge, this is the first solution using SCT with key-agreement that fully meets R'-2.
5.4 Conclusion

The previous work on security-context transfer on inter-provider handover [75, 111, 162, 176, 177, 186] has already been discussed in detail in Section 3.3.2 and Section 4.5. The main advantage of our history-enriched, policy-based approach over the above mentioned solutions is that our procedures allow users and providers to base their handover decisions on the history of a security context and thus protect themselves against any potential attacks arising from the use of weak security mechanisms between a mobile device and any previously serving network. Moreover, users and providers can enforce their policies with respect to the cipher suites they allow to be used after handover and can thus protect themselves against potential attacks arising from the use of weak security mechanisms after the current handover. More formally speaking, our history-enriched, policy-based handover procedures with key-derivation fully meet the requirements R*-1, R*-2, and R*-4 to R*-8, and meet R*-3 in part (see Section 4.3.5) and are thus secure against the attacks A*-1 through A*-25 identified in Chapter 4.

With the key-agreement method suggested in Section 5.3, we additionally have solved the problem of how to agree upon a master session key to be used between MD and DEST\_k after handover in a way that does not even reveal the newly established key to the handover controlling network for the HN-controlled case. This problem has previously been addressed only by Wang et al. [177] for the first-order handover case. However, as we showed in 3.3.2, the method suggested in [177] does not in fact meet R-3 (or R*-2).

Interesting directions for future research include exploring how the key-splitting approach could be used to enhance SCT for AN-controlled and SRC-controlled subsequent handover.

Another interesting topic for future work is to implement the HEPB-handover approach for any wireless access technology. In Chapter 12 we detail the HEPB-approach for the WLAN case and discuss how the approach could be implemented in this context.
Chapter 6

Extension to Inter-System Handover and Other Related Issues

In this chapter, we discuss inter-system and intra-provider handover procedures. In particular, we extend the history-enriched, policy-based approach to the inter-system case. The extension requires changes to the content of the transferred context history that result in a larger amount of policy data that needs to be pre-defined by each MD and each network. Moreover, as different technologies typically require different lengths of master session keys, we extend the context history and the security-mechanism negotiation by a key-conversion function. Upon inter-system handover, MD and DEST\(_k\) use the negotiated key-conversion function in order to convert the master session key transferred on a SCT with key derivation to the length required by the technology of DEST\(_k\). We require a clear separation between key derivation and key conversion in order to ensure that the cryptographical strength of the master session keys stays constant on subsequent handover with SCT and key-derivation.

Furthermore, we discuss how the history-enriched, policy-based approach relates to intra-provider, intra-system handover. We show that, while part of the requirements we defined in Chapter 4 for the inter-provider, intra-system handover case are obsolete on handover within a single network (R*-1, R*-5, and R*-6), others must be required within a single network as well (R*-2, R*-3, R*-4, R*-8, R*-7).

Another related issue we briefly discuss in this chapter is predicting upcoming handover by predicting a mobile device’s future location. Similarly, we discuss recent related work on updating the location of a mobile device and on routing traffic to a mobile device its point of network attachment after handover.

Outline. Inter-system handover are discussed in Section 6.1, followed by a detailed discussion on related work for this subject in Section 6.2. The relation of our work and intra-provider, intra-system handover is discussed in Section 6.3. In Section 6.4, we provide pointers to recent work on mobility prediction and in Section 6.5 to recent work on location management. The chapter ends with a summary and conclusion in Section 6.6.
6.1 Inter-System Handover

As more and more wireless access technologies evolve, handover procedures between different wireless access technologies need to be addressed. In particular, handover procedures between wireless access networks that offer wide area coverage but low data rates and wireless access networks that offer only local area coverage but high data rates are of great interest as candidates for *inter-system handover* [152]. Handover procedures from a local area network to a wide area network offer a wider area of seamless services to a user. Handover procedures in the other direction allow a user to take advantage of higher data rates while in the coverage area of a local area network. An example for handover procedures like this is the handover procedures between UMTS or any other mobile phone network and a wireless LAN technology like IEEE 802.11 [6].

Another important application of inter-system handover is handover between already established technologies and newly evolving ones. Such handover procedures allow providers of the new technology to offer seamless service to a user in a wider area than the new technology currently covers; thus, it facilitates the migration from one technology to another. An already standardized and implemented example for this type of handover procedure are the handover procedures between GSM and UMTS [11].

Upon inter-system handover, MD changes the access technology when switching from one network access point to another. MD is equipped with several wireless interfaces and switches from sending and receiving user traffic over one interface to sending and receiving data traffic over another interface. An inter-system handover can take place between different networks of one network provider or between different network providers. However, it turns out that the intra-provider case can be treated as a special sub-case of the inter-provider case and does not need to be detailed separately.

We differentiate between two types of inter-system handover here: *vertical* and *horizontal* inter-system handover. A vertical inter-system handover takes place if the cells of two networks of different technologies overlay each other as illustrated in Figure 6.1. A horizontal

![Figure 6.1: Vertical Handover](image1)

![Figure 6.2: Horizontal Handover](image2)

inter-system handover takes place if the cells of two networks of different technologies overlap in a border area but do not overlay each other. The typical cell topology of a horizontal inter-provider handover is illustrated in Figure 6.2.
The term vertical handover originates from the vision of overlay networks that were first described in [104]. In an overlay network, different wireless technologies with increasing coverage overlay each other such that in local areas, several technologies are available for a user. A vertical handover is then a handover from a network with local coverage to a network with wider coverage or vice-versa. Some authors [184, 176, 164] use the term vertical synonymously with the term inter-system handover and refer to horizontal handover as handover within the same technology. However, inter-system handover between wireless access technologies that overlay each other are not the only form of inter-system handover. As the handover procedures between GSM and UMTS show, inter-system handover between wireless technologies that offer similar coverage areas can be meaningful as well. Although in the early stage of UMTS deployment GSM networks will overlay UMTS islands, providers can be expected to shut down GSM services in urban areas at some point. Handover between UMTS and GSM will then not be vertical, but horizontal: from UMTS islands to GSM coverage in less populated areas.

We therefore use the term *vertical* inter-system handover to denote a handover between overlaying networks and the term *horizontal* inter-system handover to denote a handover between non-overlaying networks of different technologies. The main difference between vertical and horizontal inter-system handover is that in the vertical case, handover from a wider area network to a local area network is less time-critical than handover in the other direction, as the wider area network is available throughout the coverage of the local area network.

In the following two sections, we first discuss horizontal inter-system handover and then describe vertical inter-system handover.

### 6.1.1 Horizontal Inter-System Handover

Horizontal inter-system handover procedures can be modeled by the same procedures as inter-provider handover. We therefore refer to the procedures illustrated in Figure 3.5 for the network-initiated case and to Figure 3.8 and Figure 3.9 for the mobile-initiated case. It is, however, important to note that in the inter-system case, MD and SRC\(_k\) and MD and DEST\(_k\) communicate using different wireless access technologies.

During pre-registration, MD and its home provider either establish one set of credentials that can be used for authentication upon roaming to different networks regardless of the technology used or several sets of credentials, each of which can be used upon roaming to a network that supports a particular technology. Roaming across different technologies has already been described in detail in Section 2.3.

Before a first-order handover, MD established connection with some AN supporting a technology \(T_0\). Subsequently, MD was handed over from AN to some DEST\(_1\) supporting a technology \(T_1\), from DEST\(_1\) to DEST\(_2\) supporting a technology \(T_2\), and so on. Upon a \(k\)-th-order handover, MD is connected to some network SRC\(_k\) of technology \(T_{k-1}\) and is to be handed over to some DEST\(_k\) supporting a technology \(T_k\). As in the inter-provider intra-system case, subsequent handover can be controlled by HN, AN, or SRC.
The security challenge arising from inter-system handover consists of the same trade-off between efficiency and security we described in Section 3.2. An additional efficiency challenge, however, arises from the battery lifetime. Instead of keeping all wireless interfaces up at all times, wireless interfaces should only be brought up right before a handover to the interface’s technology will take place. For further reading on this topic, we refer to [164].

The security solutions for inter-system handover can be divided into solutions using a full new authentication and SCT-based method. In the following sections we generalize the inter-provider security solutions we have discussed in Section 3.2 to the inter-system case. In particular, we generalize the history-enriched, policy-based approach to the inter-system case.

6.1.1.1 Pre-Authentication over NAP\textsubscript{DEST},

In the inter-provider case we have discussed (see Section 3.2.1) the fact that pre-authentication via NAP\textsubscript{DEST} is only possible if the technology supports soft handover procedures. Similarly, in case of inter-system handover, pre-authentication via NAP\textsubscript{DEST} can only be used if MD can send and receive on the two involved wireless interfaces at the same time. As long as the two interfaces operate in different frequency bands, this is not a problem. Nevertheless, problems can arise in case the wireless interfaces operate in the same frequency band. For example, the two wireless local area technologies Bluetooth and WLAN are well-known to interfere with each other [78].

If the two respective technologies do not interfere, pre-authentication may still not be usable. In order to be able to use pre-authentication successfully, MD has to be within the intersection of two cells, one of each technology, from the handover detection until the successful handover completion. In particular, MD has to be in this cell intersection for the whole pre-authentication time. Whether or not pre-authentication can be used thus also depends on the sizes of cell intersections as well as the expected velocity of MDs (see Section 3.2.1 for details).

In summary, pre-authentication is a good choice to secure horizontal inter-system handover, in case the involved technologies do not interfere with each other and users can be expected to move with low velocity.

6.1.1.2 Pre-Authentication over NAP\textsubscript{SRC},

As argued in the inter-provider, intra-system case in Section 3.2.2, the problem of pre-authentication via NAP\textsubscript{SRC} is that the authentication traffic between MD and DEST\textsubscript{k} has to be tunneled through SRC\textsubscript{k}. This can be particularly difficult if the authentication and key-agreement protocols the technology DEST\textsubscript{k} uses are implemented below a common network layer. Consequently, whether or not pre-authentication over NAP\textsubscript{SRC} is a viable solution depends on the respective wireless technology.
6.1.1.3 History-Enriched, Policy-Based SCT with Key Derivation

Security-context transfers with key derivation can be used to accelerate inter-system handover in a similar way as in the inter-provider case. However, difficulties arise from the fact that the security suites specified for the technologies involved may differ from each other. Additionally, the length of the master session key required for each technology may be different. This makes two changes to our history-enriched, policy-based approach necessary. For once, the security suite histories now include cipher suites specified for different technologies and second, during the security-mechanism negotiation, MD, HCN, and DEST$_k$ have to agree upon a key-derivation function by which DEST$_k$ and MD convert the transferred master session key $K_k$ to the right length.

In the rest of this section, we discuss these changes to the history-enriched, policy-based SCT presented in Section 3.2.3 (see also Figure 3.13) in more detail. Note that we require a $k$-th inter-system handover procedure to meet the requirements defined for the inter-provider case in Section 4.3.5 and to meet one additional requirement that will be explained in the following.

We denote the set of roaming security suites specified for the technology of AN with $(R)SS_0$ and the set of cipher suites specified for the technology of DEST$_i$ with $CS_i$. We define $h$ as the maximum number of subsequent handover, that can be accommodated in the HEPB-based approach.\footnote{How large $h$ can be depends on the technologies used and has to be determined for each case separately.} The length of a master session key used in AN’s technology is denoted by $l_0$ and the length of a master session key used in DEST$_i$ ($1 \leq i \leq h$) is denoted by $l_i$. Furthermore, we assume that for each pair of length $(l_i, l_j)$, there is a set $KC_{i,j}$ of key-conversion functions $kc$ that converts an input of length $i$ to an output of length $j$. We assume that these key-conversion functions are known to all MDs and all networks.

Intuitively a key-conversion function $kc \in KC_{i,j}$ with $i \leq j$ should convert a key $K$ of length $l_i$ to a key $L$ of length $l_j$ such that $L$ is as random as $K$. Similarly a key-conversion function $kc \in KC_{i,j}$ with $j \leq i$ should convert a key $L$ of length $l_i$ to a key $K$ of length $l_j$ such that $L$ is as random as possible. We do not precisely define the indistinguishability arguments necessary to formally define the intuitively described properties of key-conversion functions. We do however suggest candidates for key-conversion functions with the desired properties:

An example for a key-conversion function for a pair of length $(l_i, l_j)$ with $l_i = \frac{1}{2}l_j$ is

$$kc_<(K) = K_1 \oplus K_2 \| K_1 \| K_2 \| K_1 \oplus K_2,$$

where $K = K_1 \| K_2$ and $K_1$ and $K_2$ are both of length $\frac{1}{2}l_i$.

An example for a key-conversion function for a pair of length $(l_i, l_j)$ with $l_i = 2l_j$ is

$$kc_>(L) = L_1 \oplus L_2,$$

where $L = L_1 \| L_2$ and $L_1$ and $L_2$ are both of bit-length $l_j$.

It is important to note that applying $kc_<$ to $K$ and then applying $kc_>$ to the output reconstructs $K$. Intuitively, this implies that $kc_<$ and $kc_>$ exhibit the desired properties: if
the output of $kc_<$ would be less random then a recovery of $K$ by such an easy to compute function as $kc_>$ would not be possible.

The length of master keys typically is $2^x$ for some $x \in \mathbb{N}$. Moreover, the composition of two key-conversion functions that exhibit the desired property seem to exhibit the desired property as well. From the two above examples we can therefore construct key-conversion functions for arbitrary pairs of key length $(l_i, l_j)$ where $l_i = 2^r l_j$ for some $r \in \mathbb{Z}$.\footnote{For example, if we want to convert a key $K$ of length $l_i$ to a key $K^{**}$ of length $l_j = 4/l_i$, we can use $kc_>$ to convert $K$ to a key $K^*$ of length $1/2l_i$ and then use $kc_>$ again to convert $K^*$ to a key $K^{**}$ of the desired length.}

**Context History.** The context history on a $k$-th-order inter-system handover is
\[
\text{history}_{k-1} = ((r)ss_{0}, kd_{0}, [kd_1, \ldots, kd_{k-1}], kc_1, \ldots, kc_{k-1}, cs_1, \ldots, cs_{k-1}, T_{k-1})
\]
where $cs_i \in CS_i$, $kc_i \in KC^{i-1,k}$. $T_{k-1}$ is defined as in the inter-provider case.

**Security Policies.** HCN, MD, and DEST$_k$ pre-define policies with respect to the cipher suites and key-conversion functions $kc \in KC^{k-1,k}$ they allow to be used after a $k$-th-order handover to a technology $T_k$, given any security-suite history $ssh_{k-1}$. They express these policies by defining subsets $CS_{HCN|ssh_{k-1}}$, $CS_{MD|ssh_{k-1}}$, and $CS_{DEST_k|ssh_{k-1}}$ of $CS_k \times KC^{k-1,k}$.

HCN, MD, and DEST$_k$ have to pre-define such sets for each $ssh_{k-1}$ and each $(1 \leq k \leq h)$, where $h$ is some fixed maximum number of subsequent inter-system handover.

Note that in the intra-provider case SRC$_k$ and DEST$_k$ are operated by the same provider and handover procedures are SRC-controlled. This facilitates the storage and processing of policy-related data as it can be handled by a single central component. In the intra-provider case, the security mechanism negotiation phases between the networks can consequently be omitted.

**Handover Agreement.** As in the inter-provider case, each destination network that has a handover agreement with HCN commits to a superset $CS^{*}_{DEST|ssh_{k-1}}$ for each $(1 \leq k \leq h)$ and each security-suite history $ssh_{k-1}$. DEST also commits to an upper bound $u_{DEST}$ on its threshold of the maximal lifetime of an initial security context.

**Key Derivation, Key Conversion, and Security Context.** During the last authentication and key agreement, AN and MD agreed upon the initial master key $K_0$ of some bit-length $l_0$. The bit-length of $K_0$ depends on the key-agreement protocol used. The key derivation depends on who controls the handover.

**HN and AN-Controlled Case.** HN (AN) derives $K_k$ from $K_0$ with the help of a (length-preserving) pre-image resistant hash function $kd_0$ that takes $K_0$ and the identity of DEST$_k$ as input.
6.1 Inter-System Handover 175

**SRC-Controlled Case.** \( \text{SRC}_k \) generates the master key \( K_k \) from \( K_{k-1} \) by means of a length preserving, pre-image-resistant hash function \( kd_{k-1} \) that additionally takes the identity of \( \text{DEST}_k \) as input. \( kd_{k-1} \) is negotiated between \( \text{SRC}_{k-1} \), \( \text{SRC}_k \), and MD during the \((k-1)\)-st-order handover \((k \leq 2)\) or during the initial security-mechanism negotiation \((k = 1)\).

For any type of handover control, the security context transferred on a \( k \)-th-order handover is

\[
S_k := (K_k, \text{history}_{k-1}, Tr_{\text{HCN}}).
\]

The procedure itself, as well as the security-mechanism negotiation, can be implemented in the same way as on inter-provider handover. However, as part of any of the security-mechanism negotiation methods described in Section 5.1, MD, \( \text{DEST}_k \), and HCN now additionally negotiate a key-conversion function \( kc_k \). Upon receipt of \( K_k \) in a handover request from HCN, \( \text{DEST}_k \) uses the negotiated key-conversion function \( kc_k \) to convert the received master key \( K_k \) to a key \( K^*_k \) of the appropriate length \( l_k \) of \( \text{DEST}_k \)'s technology. MD uses the key-derivation function \( kd_{k-1} \) negotiated during the \((k-1)\)-st-order handover to derive \( K_k \) and then uses the key-conversion function \( kc_k \) negotiated during the \( k \)-th-order handover to convert \( K_k \) into \( K^*_k \).

We require a clear separation between key derivation and key conversion. The use of the pre-image resistant hash function \( kd_{k-1} \) meets \( \text{R}^*-2 \) and part of \( \text{R}^*-3 \) and guarantees that all subsequently used master keys have the same length as the original master key \( K_0 \). \( kc_k \) converts the received master key to the appropriate length. By clearly separating these two functionalities, we aim to avoid that on subsequent handover, the randomness of the master key is reduced with each subsequent handover and once reduced, stays at the reduced level. Due to the separation, all transferred keys have the same length and are are as random as the used hash function allows and even if on a \( k \)-th-order handover the key length has to be reduced, the key transferred on the \((k+1)\)-st-order handover has the same length as the initial master session key \( K_0 \).

Note that while in the HN or AN-controlled case, the key-conversion function could be implemented in HCN. Thus, it is crucial to implement \( kc \) in \( \text{DEST}_k \) in the SRC-controlled case; otherwise, \( \text{DEST}_k = \text{SRC}_{k+1} \) may have to derive \( K_{k+1} \) from a \( K^*k \) which may be shorter than \( K_k \).

6.1.1.4 History-Enriched, Policy-Based SCT with Key Agreement

SCT with key agreement on inter-system handover requires the same changes to the SCT-based procedure presented in Section 5.2 as in the case of SCT with key derivation.

If new keys are agreed upon via the source network or multiple initial keys are generated during the initial key agreement, \( \text{DEST}_k \) may have to convert the received master key to the correct length. An important difference to the inter-provider case is that the master key \( K_k \) agreed upon with \( \text{SRC}_k \) or AN may be shorter then a master key agreed upon between MD and \( \text{DEST}_k \) upon roaming would be. The keying material used in \( \text{DEST}_k \) after an
inter-system handover with a key agreement of this type may thus be weaker (or stronger) than upon roaming.

The key-splitting approach can easily be adapted to the inter-system case if all technologies support public-key-based key agreement.

### 6.1.2 Vertical Inter-System Handover

On vertical inter-system handover, the technology with wide coverage and large cells typically offers low data rates, while the technology with local coverage and small cells offers higher data rates. A handover from the local area technology to the wide area technology is called a handover in the \textit{upward} direction. A handover in the other direction is called a handover in the \textit{downward} direction.

Vertical inter-system handover can be modeled by the same procedures as horizontal inter-system handover. In addition, the security challenge on vertical handover in the upward direction is the same as in the horizontal inter-system handover case. The main difference between vertical and horizontal handover is that handover procedures in the downward direction are less time-critical as horizontal inter-system handover. This is due to the fact that the wide area network is available to MD even after it moved into the range of a local area network. The MD can thus stay connected to the wide area network as long as a handover to the local area network is prepared regardless of any cell intersection sizes.

For a vertical inter-system handover in the upward direction, pre-authentication over the new $\text{NAP}_{\text{DEST}, k}$ is the method of choice as security solution. Nevertheless, the two involved technologies are required not to interfere with each other such that both wireless interfaces can be active at the same time. Moreover, the authentication has to be finished in time before MD moves out of range of $\text{NAP}_{\text{SRC}, k}$. In the downward direction, pre-authentication is the method of choice in the same case. The only additional difficulty here is to avoid unnecessary authentications, as MD is always in sight of the wide area technology. This problem is closely related to the problem of finding the right point in time to activate the wider area interface while connected to the local area network. An easy but power-consuming solution to this problem obviously is to keep both interfaces up at all time and keep MD permanently authenticated to the wide area network.

If SCT is to be used for any direction of vertical handover, this can be done in exactly the same way as described in the horizontal case. If SCT is to be used only in the upward direction, any handover to a local area network generates new keys and thus restarts the key cycle freshly. This reduces the risk of subsequent transfers of compromised keys.

### 6.2 Related Work on Inter-System Handover

Vertical handover procedures have been specified for handover between CDMA2000 and WLAN, GSM and WLAN, and UMTS and WLAN.

Molloy et al. \cite{130} and Buddhikot et al. \cite{43} suggest using pre-authentication upon handover from CDMA 2000 to WLAN. The procedures they suggest are mobile-initiated
and use Mobile IP for mobility management. While Molloy et al. assume MD to be pre-registered with a CDMA 2000 provider and a WLAN provider, Buddhikot et al. assume MD to be pre-registered and share credentials with only one provider.

As opposed to [130, 43], Parikh et al. [140] study handover in the upward direction only. As opposed to our model, Parikh et al. assume MDs to have two independent subscriptions, and assume no a-priori trust relationship between the inter-operating network. The authors suggest using a pro-active authentication with the CDMA 2000 network while MD is still connected to WLAN and before the WLAN signal falls below a certain ratio. The point in time at which a pro-active authentication takes place is triggered by a location-based handover prediction mechanism.

Inter-system handover procedures between UMTS and WLAN are currently being standardized by 3GPP. Many current works [157, 44, 65, 30, 186, 97, 144] summarize this standardization process.

Salkintzis et al. [157] introduce the notion of tight coupling, as well as the loose coupling between a WLAN and a UMTS network. In tight coupling, a WLAN network acts as a cell in a GPRS network, all traffic is GPRS traffic encapsulated in WLAN traffic, and the original GPRS authentication and key agreement is used regardless of whether a mobile device requests access over a WLAN network or a regular GPRS cell. In the loose-coupling approach, the GPRS authentication is used to implement an EAP-Method.

Buddhikot et al. [44] discuss the infrastructure for loosely coupled and tightly coupled solutions. They show how their loosely coupled solution for CDMA-WLAN integration can be generalized to WLAN-UMTS inter-working and they analyze the performance of their approach. Security-related issues are not emphasized in this work. However, the authors assume that each mobile device has two independent subscriptions including two pairs of credentials used to authenticate to the respective technology.

The 3GPP standard [12] for now only specifies the loosely coupled solution and advises the usage of EAP-AKA or EAP-SIM upon roaming to a WLAN. Handover procedures are not yet specified.

The inter-system handover suggestion of Zhang et al. [186] has already been discussed in connection with our threat analysis in Chapter 4.

Prasithsangree et al. [144] describe an authentication mechanism for loosely coupled UMTS-WLAN inter-operation that integrates the initialization of accounting with the authentication protocol used on accessing the WLAN. The authentication protocol used is PKI-based and assumes that each HN acts as CA for its pre-registered users. FN authenticates MD by verifying its certificate with the help of HN’s public signature-verification key. However, the authors do not address the problem of certificate-revocation status checking. Moreover, it is unclear how MD authenticates FN.

The most well-known example of horizontal inter-system handover are the handover procedures specified by 3GPP for handover between UMTS and GSM [11]. We will study these procedures in detail in Chapter 9.

Alsenmyr et al. [17] study horizontal inter-system handover between local CDMA 2000 coverage and global GSM coverage during the transition from 2G to 3G. However, security
The technology-independent work on inter-system handover [162, 176, 75, 184] has already been discussed in the context of the Related Work section in our threat analysis (see Section 4.5).

The first work on vertical inter-system handover was presented by Stemm et al. [164]. The authors suggest a mobile-initiated soft vertical handover procedure between different layers of an overlay network. Interfaces of a mobile device on higher layers are set asleep and only wake up and stay in idle mode triggered by unspecified location-based events. Interfaces on one lower layer stay idle all the time to detect “better quality” connectivity. For intra-system horizontal handover, the authors use buffering on several NAPs. A certain set of buffering NAPs is selected by MD. One of them is selected as the forwarding one. If the signal of the currently forwarding NAP falls below a certain threshold, MD commands one of the other buffering NAPs to forward and then commands the old NAP to stop forwarding. A vertical handover in the upward direction is initiated if a certain number of beacons from the current network were not received. A downward vertical handover is initiated if several beacons are received from the next lower layer.

Policy-based handover has previously been suggested by Wang et al. [175]. The authors describe a handover procedure that allows users to express policies specifying the “best” currently available wireless access technology. As a result, candidate networks are selected by finding an optimal point on a user-specified trade-off curve between the cost of the network access, the available bandwidth and the expected power consumption. However, policies with respect to security mechanisms or protection levels are not taken into account.

### 6.3 Intra-Provider, Intra-System Handover

Intra-provider handover within a single wireless access network are in use in every state-of-the-art mobile telecommunications technology, e.g., GSM and UMTS, and are lately also standardized for handover within an IEEE 802.11 WLAN. [60, 11, 91, 93]. Another WLAN-specific security solutions for intra-provider handover is the work of Zeadally et al. [185] that is based on the broadcast authentication protocol TESLA. An overview on current WLAN-specific intra-provider security solutions is provided in [27]. The security solutions used upon handover within GSM and UMTS will be detailed in Chapter 9.

Intra-provider handover within the same technology imply the security challenge that MD and a destination EIPE, if different from the source EIPE, of the handover have to be assured of each other’s authorization, establish cryptographic keys, and negotiate the cipher suite to use after handover.

The history-enriched, policy-based approach can be adapted to the intra-provider case by replacing the entities HCN, SRC_k, and DEST_k with the corresponding network components within the network of a single provider. However, not all of the requirements defined for the inter-provider case seem appropriate for intra-provider handover as well. In particular,
as on each authentication within the same network, a mobile device is authenticated in the same way, regardless of the NAP with which it is associated, such that a pre-authentication seems obsolete. Moreover, as the EIPE in charge before handover and the EIPE after handover belong to the same network and are operated by the same network provider, they will typically support the same cipher suites. Consequently, on intra-provider handover, MD and the network can typically maintain and reuse the initial security context after each handover. The problems of basing the handover decision on previously used cipher suites (R*-1), negotiating a cipher suite to use after handover (R*-5), and protecting this cipher-suite negotiation against bidding-down attacks (R*-6) thus become obsolete.

However, the requirements R*-2, R*-3, R*-4, R*-7, and R*-8 stay valid in the intra-provider case and can be met in the same way as detailed for the inter-provider case.

6.4 Mobility Prediction and Handover

Recent analyses of the intra-provider handover procedures in 802.11 WLANs have shown that the scanning for candidate destination NAPs has the largest impact on the overall handover delay [127, 173]. The authors of [160, 159] suggest reducing the scanning part of the delay by providing MD with a list of channels to scan depending on its current location. In [128, 129], this list of channels is determined using neighbor graves to describe the topology of an IEEE WLAN. In [138], frequent handover regions are used to predict the movement of a data and the most probable destination NAP of the next handover. In addition to reducing the scanning part of the handover delay, predicting the next destination NAP has the advantage that the security context can be transferred to the destination NAP pro-actively, before a handover reason occurs. During the handover execution, MD then only has to switch the link-layer connection, while the security context is already in place.

In joint work with K. Kastell, A. Fernandez-Pello, D. Perez and R. Jakoby, we have adapted this approach to the GSM/UMTS case. In particular, we show in [102] that the handover preparation time on any type of handover within and between GSM and UMTS can be reduced by at least 480 ms by predicting the destination NAP based on a simple triangulation of a user’s location.

6.5 Location Management

Mobility management in wireless access networks can be divided into roaming and handover procedures on one hand, and location management on the other hand [147]. Throughout this thesis, we concentrate on roaming and handover procedures. However, in this section, we provide some pointers to recent work on location management.

Location management consists of two stages: location updates in which new location information on MD is registered and traffic delivery in which MD is localized, and incoming

\footnote{The overall handover delay is measured from the detection of a handover reason until the first frame of user data is received over the new link connection.}
and outgoing traffic is routed to MD’s current point of network attachment.

The most commonly discussed location-management protocol for IP-based mobility is Mobil IP [142], which allows a mobile device to maintain its IP address while roaming to different domains. The delay introduced by Mobile IP to inter-domain handover has, for example, been studied in [172]. Although these results are promising, Mobile IP seems to produce a large overhead when used to provide local mobility to large amounts of mobile users. Consequently, many alternative mobility-management protocols have been suggested. Most of these protocols are optimized for local mobility and can be used to supplement Mobile-IP-based macro-mobility management. Examples for such supplementary suggestions are Hierarchical-IP [161], Cellular-IP [171], HAWAII [148], and TeleMIP [50]. An overview of these protocols can be found in [151].

6.6 Conclusion

In this chapter, we have extended our discussion on inter-provider handover to the vertical and horizontal inter-system handover cases. We briefly discussed to what extent the history-enriched, policy-based approach is meaningful to intra-provider handover within the same technology. Moreover, we have provided pointers to handover-related issues, such as mobility prediction and location management, that are considered out of the scope of this work.

The main contribution of this chapter is the extension of the history-enriched, policy-based handover approach to the inter-system case. To accommodate inter-system handover with key derivation, we extend the context history to contain information on the technology-specific security suites used between MD and any previously serving networks before a \( k \)-th-order handover. As a consequence, MD, HCN, and DEST\(_k\) have to pre-define policies for each of these new context histories and each order of subsequent handover. Moreover, as different technologies typically require master session keys of different length, we add a key-conversion function to any \( k \)-th-order handover procedure. This key-conversion function converts the transferred master key (derived as in the inter-provider case) to the appropriate length required by DEST\(_k\)’s technology. We require that a key-conversion function shall maintain as much randomness as possible such that the input key is cryptographically as strong as the output key. We provide simple examples for key-conversion functions with this property, which can be used for conversions between arbitrary key lengths. However, we do not formally define the above desired property of a key-conversion function and consequently do also not prove that the two suggested functions have the desired property. This is an interesting topic that requires further investigation.

Throughout this work, we assume that all subsequent handover procedures are of the same control type. However, in the inter-system case, this is a somewhat restricting assumption. An interesting direction of future research may be studying the security impact of subsequent handover with changing control types, which will in particular require more complex assumptions about the trust relations between providers.
Part III

Handover and Roaming within and between GSM and UMTS
PART III IN THE GENERAL CONTEXT
In this part, we describe and analyze the security architecture, roaming and handover procedures within and between GSM and UMTS networks. Chapter 7 describes roaming between GSM networks and between UMTS networks. In Chapter 8 we describe and analyze inter-system, intra-provider as well as inter-system, inter-provider roaming between GSM and UMTS. In Chapter 9, we describe inter-system, intra-provider and inter-system, inter-provider handover procedures between GSM and UMTS. We analyze these procedures with respect to the security requirements and threats introduced in Chapter 4 and describe the impact of selected attacks against GSM.
Chapter 7

Inter-provider Roaming within GSM and UMTS

The Global System for Mobile communication (GSM) is one of the second generation (2G) mobile phone network technologies adopted as standard by the European Telecommunications Standard Institute (ETSI) in 1989. With 1.5 billion GSM subscribers in September 2005 [80], GSM is currently the most widespread technology underlying mobile phone networks.

Although GSM is available in over 200 countries, GSM does not provide worldwide coverage. The main 2G mobile communication standards competing with GSM are the American IS-95, used in the US and some Asia countries, and the Japanese PDC, used in Japan and Korea. The incompatibilities of these 2G systems led to the vision of a single third generation (3G) mobile communications standard that would allow for easy worldwide roaming. The standards family IMT-2000 defined by the ITU is the result of this vision and consists of several compatible 3G standards that will allow multi-mode MDs to roam between all standards in IMT-2000. The Universal Mobile Telecommunications Standard (UMTS) standardized by 3GPP is one of the members of the IMT-2000 family and allows for handover and roaming between UMTS and GSM. As such, UMTS is designed to be the successor of GSM. The first phase of the UMTS standardization was finished in 1999. UMTS is in operation in more than 80 commercial networks in 35 countries and in September 2005 the number of UMTS subscribers worldwide exceeded 33 million [170].

In this chapter, we give a brief overview on the roaming procedures supported by the two mobile communication standards GSM and UMTS.

Outline. GSM inter-provider roaming is described in Section 7.1. We start by giving an overview on the GSM system architecture in Section 7.1.1. We then describe the GSM security model in Section 7.1.2. Intra-provider roaming in GSM is briefly discussed in Section 7.2 while UMTS inter-provider roaming is described in Section 7.3. Again, we start by describing the UMTS system architecture in Section 7.3.1 and then give an overview on the UMTS security architecture in Section 7.3.2. We finally describe UMTS intra-provider roaming in Section 7.4.
7.1 GSM Inter-Provider Roaming

7.1.1 System Model

In a GSM network, a mobile device is connected to a *visited network*\(^1\) via a radio link to a particular Base Transceiver Station (BTS). Multiple BTSs are connected to a Base Station Controller (BSC) and multiple BSCs are controlled by a Mobile Switching Center (MSC). A BSC, together with the BTSs connected to it, is also referred to as a Base Station Subsystem (BSS) or GSM EDGE Radio Access Network (GERAN). Each MSC has access to a Visitor Location Register (VLR) that keeps track of the location of all Mobile Devices (MDs) currently connected to the visited network. The Home Location Register (HLR) in HN keeps track of the location of all MDs that are pre-registered with HN. The Authentication Center (AuC) stores all security-related information of all pre-registered users.\(^2\) Figure 7.1 describes the GSM network architecture. Moreover, it illustrates the keys and security mechanisms stored by the network components, which are described in detail in Section 7.1.2.

\[\text{Figure 7.1: System Model and Storage of Security Information}\]

7.1.2 Security Model

GSM supports mobile device authentication as well as encryption of the air interface between a mobile device and a network access point (BTS). GSM does not support network

\(^1\)Its home network or a foreign network that has a roaming agreement with MD’s home network.

\(^2\)In our security model we used the term Security Center (SC) instead of AuC.
authentication and integrity protection. A detailed description of all GSM security features can be found in [62]. In the following overview, we emphasize the authentication and key agreement, as well as the security-mechanism negotiation on GSM inter-provider roaming.

7.1.2.1 Registration Process

Every GSM user (also referred to as a GSM subscriber) registers for a dedicated home network that is operated by his home provider. During registration, the home provider allocates an International Mobile Subscriber Identity (IMSI) for the user and generates a long-term secret key $K_i$ of 128 bits. The credential pair (IMSI, $K_i$) is stored in HN’s AuC as well as on a smart card, the Subscriber Identity Module (SIM). This smart card is handed out to the user, who plugs it into his MD. Aside from the credential pair, the SIM also contains two provider-specific algorithms A3 and A8 that are—as will be explained in more detail further below—used for authentication and key generation respectively. During registration, the user furthermore registers for a certain geographical roaming region.

7.1.2.2 Authentication and Key Agreement

The GSM standard does not allow for mutual authentication on inter-provider roaming. Upon roaming, FN is assured of MD’s authorization to roam to FN but MD does not authenticate FN. This enables a network impersonation attack against MD, which we will briefly discuss in Section 9.4.1.2.

The roaming authentication and key-agreement protocols are implemented together and require HN’s interaction on the first authentication of MD to FN. An HN set number of subsequent authentications of MD to FN do not then require HN’s interaction. On the first authentication, FN requests authentication data from HN. HN provides FN with one or more authentication vector(s), each consisting of a random challenge $RAND_G$, an authentication response $RES_G$, and an encryption key $Kc$. $RAND_G$ is randomly chosen in HN’s AuC. $RES_G$ is generated by AuC from $RAND_G$ and $K_i$ with the help of some (HN-specific) algorithm A3. Similarly, $Kc$ is generated by AuC from $RAND_G$ and $K_i$ with help of some (HN specific) algorithm A8.

FN presents MD with $RAND_G$. MD’s SIM card generates $RES_G^*$ and $Kc$ from $RAND_G$ and $K_i$ with help of A3 respectively A8 and sends $RES_G^*$ back to FN. If $RES_G^*$ equals $RES_G$, FN has successfully authenticated MD.

A3 and A8 are HN-specific algorithms. GSM can thus theoretically support as many authentication and key-agreement protocols as there are GSM operators. In fact, only a few A3 and A8 implementations are in use. As FN does not have knowledge of the long-term secret key $K_i$, HN has to generate the encryption key $Kc$. A secure channel between HN and FN is needed to secure the transfer of authentication vectors.

In summary, the GSM inter-provider roaming authentication and key-agreement protocol(s) are of Type3 (see Section 2.1.5), require a secure channel between HN and FN, and are secret-key-based.

In GSM, the master key is identical to the encryption key. A key-establishment process \( ke \) (see Definition 1.2.9) is not used.

### 7.1.2.3 Encryption

The GSM standard currently specifies four encryption mechanisms, namely the stream ciphers A5/0 (no encryption), A5/1 (standard encryption), A5/2 (weaker version of A5/1) and A5/3 (similar to the KASUMI encryption mechanism used in UMTS [8]).\(^4\) The encryption mechanisms are implemented on MD\(^5\) and in all BTSs. They are used to protect the confidentiality of all data traffic and some specific control messages between MD and BTS. As opposed to A3 and A8, the mechanism of the A5 family are standardized and not provider-specific. Each provider may choose a set of encryption mechanisms it supports. This choice may also depend on regional restrictions on the use of cryptographic techniques.

### 7.1.2.4 Security Mechanism Negotiation and Policies

As each SIM card supports exactly one pair of A3 and A8 algorithms, no authentication and key-agreement protocols have to be negotiated during connection establishment. As GSM does not support integrity protection and does not use any key-establishment process between MD and BTS, each cipher suite in GSM consists of one encryption mechanism only. MD and FN negotiate the encryption mechanism on roaming without HN’s interaction. Upon connection establishment with FN, MD sends its security capabilities, a list of A5 algorithms MD supports, to FN. By means of the standard, MD is mandated to support A5/0, A5/1, and A5/2. FN is required to drop the connection if it receives security capabilities from MD that do not include the mandatory algorithms. In particular, the definition of mandatory algorithms protects the security-mechanism negotiation against a bidding-down attack below the protection level offered by A5/1.\(^6\) FN chooses one of the algorithms MD supports and acknowledges its choice to MD in a GSM security mode command message. It is interesting to note that HN has no influence on FN’s choice of the encryption algorithm. In particular, HN cannot forbid the use of A5/0 = no encryption or the weak encryption mechanism A5/2. Similarly, MD cannot enforce the use of the stronger algorithms A5/1 or A5/3.

### 7.1.2.5 Anonymity

To protect the confidentiality of a GSM subscriber’s identity, the IMSI is sent in the clear on the air interface as rarely as possible. Instead, a Temporary Mobile Subscriber Identity TMSI is allocated by FN, to which MD last presented its IMSI. On the next connection establishment, MD presents its TMSI to the network. The network tries to resolve the

---

\(^4\)The GSM standard can accommodate seven A5 mechanisms of which only the first four are currently specified.

\(^5\)As opposed to A3 and A8 that are implemented on the SIM card [9].

\(^6\)A5/3 is generally rated the strongest cipher in the A5 family, followed by A5/1. A5/2 is de facto broken [28] and A5/1 has been shown to be weak [33, 34].
TMSI and obtain the IMSI from the old VLR. If the network cannot resolve the TMSI, it requests MD to present its IMSI in the clear. Upon allocation, a new TMSI is encrypted.

### 7.1.2.6 Summary and Connection Establishment

Figure 7.2 summarizes authentication and key agreement, as well as anonymity protection, security-mechanism negotiation, and encryption, on inter-provider roaming in GSM by detailing the connection establishment between MD and FN on roaming.

In order to facilitate future reference to the GSM connection-establishment process, Figure 7.2 is divided into four parts. In the first part (GSM I), MD presents its identity to FN. In the second part (GSM II), FN obtains one or more authentication vectors from HN.\(^7\) In the third part (GSM III), FN authenticates MD based on an authentication vector. Finally, in GSM IV the encryption key is transferred to BTS and MD is informed which encryption mechanism to use.

### 7.2 GSM Intra-Provider Roaming

Inter-provider roaming from one GSM network provider to another always causes a new roaming authentication between the network and MD. This is not the case if a MD is in idle mode\(^8\) and moves within the home or a foreign network it roamed to. In this case, MD is authenticated once it roams to the visited network, or once the provider of the visited network requests a new authentication.\(^9\) When moving out of the range of the currently serving BTS and into the range of a new one, the encryption key \(K_c\) negotiated during the last authentication is transferred from the MSC to the new BTS. If MD roams beyond the control of its currently serving MSC, the MSC first transfers \(K_c\) to the new MSC and the new MSC subsequently transfers \(K_c\) to MD. MD is indirectly re-authenticated by the new BTS, as only the legitimate MD has knowledge of the currently used \(K_c\). As BTS cannot distinguish between correctly and incorrectly encrypted data traffic and no integrity protection is used in GSM, BTS cannot detect impersonated MDs by means of this re-authentication method. However, trying to impersonate MD to the new BTS is useless for an attacker without knowledge of \(K_c\), as long as the new BTS enables encryption. In this context, it is interesting to note that if encryption was disabled between MD and the source BTS, it would not be enabled after roaming to the new BTS. Consequently, the fast re-authentication in idle mode opens up to an impersonation attack. The GSM security standard [62] does not specify how the A5 algorithm selection is handled in case of roaming.

\(^7\)Note that this phase does not take place if FN still stores unused authentication vectors from prior authentications.

\(^8\)In accordance to Section 2.4.2 we differentiate between intra-provider roaming and intra-provider handover. A handover takes place if a user moves during an ongoing connection while a roaming procedure takes place while the user is in idle mode, i.e., currently has no on-going connections.

\(^9\)E.g., on periodical location updates or in certain provider set time intervals. The standard only defines, in which situation a network may request a new authentication but does not require it to do so on any occasion.
the encryption was not disabled. In particular, it is unclear what happens if the new BTS does not support the A5 algorithm used between the source BTS and MD\textsuperscript{10}

\textsuperscript{10}Most networks can be expected to support the same algorithms on every BTS throughout the network, such that the same algorithm can be used before and after handover. However, e.g. due to a sequential upgrade of BTSs, this is not necessarily the case.
7.3 UMTS Inter-Provider Roaming

7.3.1 System Model

In a UMTS network, MD is connected to a visited network via a radio link to a particular base transceiver station, called NodeB in UMTS. Multiple NodeBs are connected to a Radio Network Controller (RNC) and multiple RNCs are controlled by a Mobile Switching Center (MSC). The RNCs, together with the NodeBs that are connected to them, are also referred to as UMTS Terrestrial Radio Access Network (UTRAN). Each MSC has access to a Visited Location Register (VLR) that keeps track of the location of all MDs currently connected to the visited network. The Home Location Register (HLR) in HN keeps track of all MDs that are pre-registered for HN. As in GSM, all security-related information regarding MD is stored in an AuC. Figure 7.3 illustrates the UMTS system components. Moreover, it shows the keys and security mechanisms stored by each component. These are explained in detail in the following section.

![Diagram of UMTS System Model and Security Mechanism Endpoints](image)

**Figure 7.3: UMTS System Model and Security Mechanism Endpoints**

7.3.2 Security Model

As opposed to GSM, the UMTS standard supports not only encryption but also integrity protection. Moreover, the authentication between MD and a visited network is mutual. The E1PE on the network side does not coincide with the network access point (NodeB). Instead, encryption and integrity protection are implemented in the RNC. An introductory overview on the UMTS security features can be found in [89]. More detailed information is provided in [133, 9].
7.3.2.1 Registration Process

Every UMTS user (also referred to as UMTS subscriber in the following) registers for a dedicated home network operated by its home provider. As in GSM, on registration HN allocates an IMSI as well as a long-term secret key $K_U$ for the user. The pair (IMSI, $K_U$) is stored on a Universal Subscriber Identity Module (USIM), as well as in HN’s AuC. The USIM also contains five key-generation functions $f_1, \ldots, f_5$. During registration, a user furthermore registers for a certain geographical roaming region.

7.3.2.2 Authentication and Key Agreement

As opposed to GSM, the UMTS standard allows for mutual authentication on inter-provider roaming. Upon authentication, FN is assured that HN authorizes MD’s roaming to FN and MD is assured of HN’s authorization of FN’s service provisioning to MD. The roaming authentication and key-agreement protocols are implemented together and require HN’s interaction on the first authentication. An unspecified number of subsequent authentications can then take place without HN’s interaction. On the first authentication, FN requests authentication data for MD from HN. HN provides FN with a provider-set number of authentication vectors. Each authentication vector consists of a random challenge $RAND_U$, an authentication response $RES_U$, an encryption key $CK$, an integrity-protection key $IK$, and an authentication token $AUTN$. $RAND_U$ is generated randomly by AuC. $RES_U$ is generated from $RAND_U$ and $K_U$ with the key-generation function $f_2$. Similarly, $CK$ and $IK$ are generated from $RAND_U$ and $K_U$ by means of $f_3$ ($CK$) and $f_4$ ($IK$). The authentication token $AUTN$ is generated from $RAND_U$, $K_U$, a sequence number SQN, and an authentication management field AMF by two functions $f_1$ and $f_5$:

$$AUTN = (\text{SQN} \oplus f_5(RAND_U, K_U), AMF || f_1(SQN, AMF, K_U))$$

(7.1)

Here, $||$ stands for the concatenation and $\oplus$ for the exclusive or operation of bit strings. Essentially, AUTN is a sequence number that is integrity-protected with a Message Authentication Code (MAC) generated with the help of the long-term secret key $K_U$.

FN presents $RAND_U$ and $AUTN$ to MD. MD’s USIM generates the response $RES_U$ as well as $IK$, $CK$ and AK. MD extracts SQN from the first part of AUTN and verifies that SQN is in the right range. MD then computes $MAC^* = f_1(SQN, AMF, K_U)$. MD discards the message if $MAC^* \neq MAC$. It is important to note that the correctness of the MAC and the fact that SQN is in the right range only proves to MD that HN has recently generated AUTN, but not whether or not FN has actually received a complete authentication vector from HN. It is only in combination with the integrity protection provided by FN that this

---

11 Used to support more than one set of key-generation functions $f_1, \ldots, f_5$.

12 The Anonymity Key (AK) aims to protect against profiling attacks enabled through the use of sequence numbers.

13 A detailed description of the construction of SQN, its re-synchronization, and the range in which it is required to be can be found in the appendix of [9].
proves FN’s authorization to MD (see Section 8.3). If MAC is correct, MD sends $RES^*_U$ back to FN and the MSC of FN verifies $RES^*_U$. FN has successfully authenticated MD if $RES^*_U$ equals $RES_U$.

The choice of the key generation functions $f_1$ to $f_5$ is left to the UMTS providers. Nevertheless, the organization of standardization 3GPP\textsuperscript{14} that standardizes UMTS specifies a sample set of functions called MILLENAGE [10]. Therefore, UMTS theoretically supports as many authentication and key-agreement protocols as there are UMTS providers. It is, however, expected that most providers will use MILLENAGE. As a consequence, FN cannot generate the data-protection keys $IK$ and $CK$. In UMTS, HN and FN thus require a secure channel between each other in order to protect the transfer of authentication vectors.

In UMTS, the master key generated during key agreement equals the pair of data-protection keys. A key-establishment process $ke$ (Definition 1.2.9) is not used in UMTS.

### 7.3.2.3 Encryption and Integrity Protection

The UMTS encryption and integrity-protection endpoint on the network side is not NodeB, but RNC. The UMTS protection thus reaches further back into the UMTS backbone network than in GSM. The encryption and integrity-protection mechanism are implemented in MD and not on USIM.

UMTS can accommodate up to 16 different encryption mechanisms. Currently only two UMTS Encryption Algorithms (UEAs) are specified, namely UEA0 (no encryption) and UEA1, a stream cipher based on the block cipher KASUMI [8].

Similarly, only one of 16 possible UMTS Integrity Algorithms (UIAs), namely UIA1, is specified in the current standard [9]. It is also based on the block cipher KASUMI [8]. As opposed to encryption, the integrity protection of dedicated control messages is mandatory.

In GSM, no mechanism to restrict the lifetime of an encryption key $K_c$ is standardized. The UMTS standard avoids this weakness in that the lifetime of the keys $IK$ and $CK$ is restricted by an HN-set threshold on how much data may at most be protected with the same key pair. This threshold is stored on the USIM and checked every time a Radio Resource Connection (RRC) is released. If a new RRC is established and the threshold was reached during the last RRC, a new authentication and key agreement are initiated. For more details on this mechanism, refer to [133].

### 7.3.2.4 Security-Mechanism Negotiation and Policies

Similar to GSM, in UMTS the visited network and MD negotiate only the encryption and the integrity-protection mechanisms to use. HN is not engaged in the negotiation. On connection establishment, MD sends its security capabilities, i.e., a list of all encryption and integrity-protection mechanisms it supports to FN. After successful authentication of MD, FN’s MSC decides which mechanisms MD and RNC are allowed to use and sends a list of allowed mechanism pairs to RNC. RNC selects one of the allowed mechanism pairs and acknowledges its choice to MD in a security mode command message. This

\textsuperscript{14}3rd Generation Partnership Project.
message also repeats the security capabilities received from MD on connection setup and is integrity protected. On receipt of this message, MD verifies the integrity protection and compares the security capabilities with the ones it originally sent. By means of this, MD can detect manipulation of its security capabilities. Furthermore, the integrity protection of this message guarantees to MD that FN is authorized by HN to offer service to MD. This is due to the fact that only HN can generate a valid integrity-protection key for a given $\text{RAND}_U$ value and presents only authorized FN’s with authentication vectors.\footnote{Below, We will show that a recently generated AUTN can be received by anyone, yet the corresponding $IK$ cannot.}

It is important to note that MD is currently mandated to support the no encryption mechanism UEA0. Consequently, neither MD nor HN can enforce encryption to be enabled. Instead, the choice of whether encryption is used or not is left to the visited network provider’s choice.

The UMTS standard makes use of the same mechanism of temporary identities to protect a UMTS subscriber’s IMSI as GSM.

### 7.3.2.5 Summary and Connection Establishment

Figure 7.4 illustrates the complete security-related procedures on connection setup of a UMTS subscriber with a UMTS network. To facilitate future references to this figure, it is divided into four parts. In the first part (UMTS I), MD presents its identity to FN. In the second phase (UMTS II), FN requests authentication vectors from HN. In the third part (UMTS III), FN authenticates MD and MD is assured that HN recently generated the received authentication token AUTN. In the fourth and final part (UMTS IV), FN chooses the encryption and integrity-protection mechanisms to use, and acknowledges them in an integrity-protected message to MD.\footnote{By this message, MD is finally assured that FN is an authorized network.}

### 7.4 UMTS Intra-Provider Roaming

Similar to intra-provider roaming in GSM, the encryption and integrity-protection keys $IK$ and $CK$ agreed upon during the last authentication and key agreement are transferred from one RNC to the next if an idle mode MD roams within a UMTS network. If the encryption was disabled by the last serving RNC, it stays disabled after roaming to the destination RNC. The UMTS security standard [9] does not specify whether the same UEA is used after roaming to the destination RNC or a new UEA is negotiated. The standard also does not address the problem of how a new UEA is selected if the destination RNC does not support the UEA that was used between the source RNC and MD.
Figure 7.4: UMTS Authentication, Key Agreement, and Security-Mechanism Negotiation
Chapter 8

Roaming Between GSM and UMTS

To facilitate the transition from GSM to UMTS, the UMTS standard allows both radio access networks UTRAN and BSS to be simultaneously operated with a single backbone network and, in particular, a single hierarchy of MSCs. For this purpose, UMTS MSCs, the so-called 3G MSCs, cannot only control UTRANs, but also GSM BSSs. In contrast, only BSSs can be connected to the original GSM MSCs, the so-called 2G MSCs. We will refer to networks in which BSSs are connected to 3G MSCs as mixed-mode networks, to networks that only operate UTRAN as UMTS networks, and to networks that only operate GSM BSSs and 2G MSCs as GSM networks.

In this chapter, we describe all inter-system roaming authentication protocols between the three above-mentioned network types. All of these procedures are specified in [11]. Here, we present a comprehensive and detailed description of each roaming case and thus clarify the standard documents. Other overviews on the roaming authentication procedures (e.g., [133, 89, 37, 145, 146]) fail to capture the essential differences between a full UMTS authentication and an authentication that is based on a UMTS-authentication vector but is carried out while MD is connected to a GSM BSS. Consequently, UMTS was to date believed to be secure against man-in-the-middle attacks.

Exploiting a weakness in the GSM/UMTS inter-system handover procedures, we present a man-in-the-middle attack on UMTS. This attack is the result of joint work with S. Wetzel and has been published in [122]. Our attack allows an intruder to eavesdrop on all mobile-initiated traffic. Possible victims to our attack are all MDs that support the UTRAN and the GSM air interface. In particular, this will be the case for most of the equipment used during the transition phase from 2G (GSM) to 3G (UMTS) technology.

The attack, as well as the countermeasures we suggested, were discussed by the standardization organization 3GPP [2]. As a result, future MDs shall be protected against our attack.

Outline. In Section 8.1 we describe the inter-system, inter-provider roaming procedures between GSM and UMTS networks. This is followed by the details on the inter-system,
intra-provider roaming procedures in a mixed-mode network, a network in which GERAN
and UTRAN coexist in Section 8.2. In Section 8.3, we present our man-in-the-middle attack
against UMTS, including the discussion of our suggested countermeasures.

8.1 GSM/UMTS Inter-System Inter-Provider Roaming

The UMTS standard allows for SIM-equipped users (users that originally subscribed to
GSM services) to roam to UMTS, as well as to mixed-mode networks. The advantage of
this roaming type is that a user can subscribe to UMTS and still keep his old SIM card.
This facilitates the process of subscribing to UMTS and saves operators from handing out
new smart cards. Vice-versa, a USIM-equipped user can roam to GSM networks and mixed-
mode networks. In the transition phase, this type of roaming is crucial for user acceptance
and satisfaction, as users at least obtain GSM services in areas that are not covered by
UMTS yet.

Combining the different types of smart cards, serving radio access network, and MSCs,
we describe six essentially different roaming authentication scenarios. These scenarios are,
if not coherent, described in [9]:

Case 1: A USIM-equipped MD roams to a UMTS network (UMTS inter-provider roaming
as described in Section 7.3.2.2).

Case 2: A SIM-equipped MD roams to a GSM network (GSM inter-provider roaming as
described in Section 7.1.2.2).

Case 3: A SIM-equipped MD roams to UMTS.

Case 4: A USIM-equipped MD roams to GSM (i.e., to a GSM BSS that is connected to a
2G MSC).

Case 5: A USIM-equipped MD roams to a GSM BSS that is connected to a 3G MSC
(mixed-mode case).

Case 6: A SIM-equipped MD to a GSM BSS that is connected to a 3G MSC (mixed-mode
case).

All six roaming scenarios are illustrated in Figure 8.1. The first two roaming cases are
the inter-provider, intra-system cases that have already been described in the last section.
The last four cases are the inter-system cases. Note that in any of the six cases the network
to which MD roams can either be operated by the home provider or by a foreign provider.
In the following sections we detail Cases 3 through 6. Note that throughout the remainder
of this part, we use use light grey to refer to GSM components and darker grey to refer to
UMTS.

In order to be able to use any of the inter-system roaming support the UMTS standard
offers, a user’s MD has to support both the GSM and the UMTS radio interface. Throughout
this work, we assume that all users are equipped with MDs of this type. In particular, MDs
of this type support GSM encryption as well as UMTS encryption and integrity protection. As UTRAN supports UMTS protection only and GERAN supports GSM encryption only, the air interface protection is determined by the radio access network.

The type of authentication and key agreement depends on the type of smart card plugged into MD, as well as the type of MSC. Each SIM and 2G MSC supports authentication based on GSM-authentication vectors only while each USIM and 3G MSC supports authentication based on both GSM-authentication vectors or UMTS authentication vectors.

### 8.1.1 A SIM-Equipped MD Roams to UMTS (Case 3)

In this case, a SIM-equipped MD connects to the network via a UMTS base station in the same way as in GSM I (Figure 7.2). Node B forwards all GSM traffic transparently. The MSC of the visited network requests a GSM-authentication vector \((RAND_c, K_c, RES_c)\) from HN, as in GSM II. The 3G MSC and MD follow GSM III to authenticate MD to FN. Again, Node B simply forwards the GSM-authentication messages. The 3G MSC sends \(RAND_c\) to MD (via Node B). MD generates the authentication response \(RES_c^*\) and the encryption key \(K_c\) from \(RAND_c\) and the long-term secret key \(K_i\). The MD sends \(RES_c^*\) back to the visited MSC, which compares \(RES_c^*\) to \(RES_c\). The authentication is deemed successful if the two values match.

After a successful GSM III, MD and MSC convert the established GSM key \(K_c\) into UMTS keys

\[
CK = c_4(K_c) = K_c || K_c
\]

\[
IK = c_5(K_c) = K_c \oplus K_c \|| K_c \| K_c \oplus K_c,
\]

where \(K_c = K_c_1 \|| K_c_2\) and \(K_c_1\) and \(K_c_2\) are 32 bits in length. MD and the visited network then follow the steps of UMTS IV (see Figure 7.4). The visited 3G MSC transfers \(IK\) and \(CK\) to RNC and RNC sends the integrity-protected security mode command message to MD. The UMTS keys \(CK\) and \(IK\) are subsequently used to encrypt and integrity-protect
the communication between RNC and MD. The complete security-related procedures on
connection setup of a GSM subscriber with a UMTS network is described in Figure 8.2.

![Diagram of communication between MD/SIM, NodeB/RNC, 3G MSC/VLR, 2G MSC/HLR]

**Figure 8.2: A SIM-Equipped MD Roams to UMTS (Case 3)**

### 8.1.2 A USIM-Equipped MD Roams to GSM Network (Case 4)

A MD equipped with a USIM connects to a GSM BTS, which is connected to a 2G MSC. Since a 2G MSC does not support UMTS authentication, a UMTS subscriber can be authenticated by a 2G MSC only if the USIM supports the conversion of UMTS-authentication vectors to GSM-authentication vectors.

MD presents its identity to the visited network as in GSM I. The visited network requests a GSM-authentication vector from the home network similar to GSM II: HN first generates a UMTS-authentication vector and then converts it into a GSM-authentication vector. The GSM-authentication challenge and the UMTS-authentication challenge are the same, i.e., $RAND_G = RAND_U$. The 32-bit GSM-authentication response $RES_G$ is generated from the 128-bit UMTS-authentication response $RES_U$ by splitting the UMTS response into four 32-bit values, such that $RES_G = RES_U_1 || RES_U_2 || RES_U_3 || RES_U_4$ and computing

$$RES_G = c_2(RES_U) = RES_U_1 \oplus RES_U_2 \oplus RES_U_3 \oplus RES_U_4$$

The GSM-encryption key is derived from the UMTS keys $IK$ and $CK$ by:

$$Kc = c_3(CK, IK) = CK_1 \oplus CK_2 \oplus IK_1 \oplus IK_2,$$  \quad (8.3)

where $CK$ and $IK$ are each split into $CK_1, CK_2, IK_1$ and $IK_2$ with a length 64 bits each, such that $CK = CK_1 || CK_2$ and $IK = IK_1 || IK_2$.

The home network forwards the GSM-authentication vector to the visited network. The visited 2G MSC in turn sends the authentication challenge to MD, which itself generates
8.1 GSM/UMTS Inter-System Inter-Provider Roaming

the GSM-authentication vector from the UMTS-authentication vector as described above. This is followed by the remainder of the standard GSM IV in which the GSM-encryption key $K_c$ is transferred to BTS and BTS sends the GSM cipher mode command to MD. MD and BTS subsequently use $K_c$ and the negotiated encryption mechanisms to protect data traffic between them. The complete procedure of a UMTS subscriber roaming to GSM is described in Figure 8.3.

![Figure 8.3: USIM-Equipped MD Roams to GSM (Case 4)](image)

8.1.3 USIM-Equipped MD Roams to a Mixed-Mode Network (Case 5)

In this case, a USIM-equipped MD connects to a GSM base station and this GSM base station is controlled by a 3G MSC. Since the 3G MSC and the USIM support UMTS I, II, and III of the UMTS connection establishment, these can be applied as described in Figure 7.4. The GSM BTS forwards the respective traffic transparently.

After completing UMTS III, MD and the visited 3G MSC convert the generated UMTS keys $IK$ and $CK$ into a GSM key $K_c$ as in Equation (8.3):

$$K_c = c_3(CK, IK)$$

MD and the GSM BSS proceed with part IV of the GSM authentication described in Figure 7.2. In particular, BTS acknowledges its choice of the encryption mechanism to MD in the GSM cipher mode command. $K_c$ and the negotiated GSM-encryption mechanism
are subsequently used to encrypt data traffic between MD and BTS. Case 5 is illustrated in Figure 8.4.

\[ K_c = c_3(IK, CK) \]

Figure 8.4: USIM-Equipped MD Roams to a Mixed-Mode Network (Case 5)

8.1.4 SIM-Equipped MD Roams to a Mixed-Mode Network (Case 6)

In case an SIM-equipped MD roams to a GSM BTS that is controlled by a 3G MSC, the 3G MSC acts in exactly the same way as on regular GSM roaming (GSM I-IV, Figure 7.2). Case 6 is illustrated in Figure 8.5.

\[ K_c = c_3(IK, CK) \]

Figure 8.5: GSM Subscriber Roaming to a Mixed-Mode Network (Case 6)

8.2 Intra-Provider Roaming within a Mixed-Mode Network

Similar to intra-provider UTRAN or GSM roaming, the UMTS standard specifies a fast re-authentication based on the currently used encryption (and integrity protection) key(s) on intra-provider roaming within a mixed-mode GSM/UMTS network.

Upon intra-provider roaming from UTRAN to GERAN, the currently used keys UMTS keys \( IK \) and \( CK \) are transferred into the GSM key \( K_c = c_3(CK, IK) \).
8.3 Man-in-the-Middle Attack on UMTS

Upon intra-provider roaming from GERAN to UTRAN, the key transfer depends on whether the GERAN controlling MSC is a 3G or a 2G MSC. A 3G MSC stores and transfers the UMTS keys, while a 2G MSC transfers the GSM key. Note that a single roaming to a 2G MSC makes the original UMTS keys unrecoverable. Consequently, if a USIM-equipped MD is authenticated by a 3G MSC and then subsequently roams to a 2G MSC, a 3G MSC and yet another 3G MSC, between the 3G MSCs, the pseudo-UMTS keys $IK^* = c_5(K_c)$ and $CK^* = c_4(K_c)$ are transferred rather then the originally generated UMTS keys $IK$ and $CK$. If a SIM-equipped MD roams within a mixed-mode network, either the pseudo-UMTS keys or the GSM key is transferred between the MSC. A conversion of a GSM key to UMTS keys and back to a GSM key recovers the original GSM key. Consequently, a transfer of $IK^*$ and $CK^*$ essentially carries the same information as a transfer of $K_c$. Figure 8.6 illustrates the key transfer of USIM and SIM-equipped MDs in a mixed-mode network.

![Figure 8.6: Intra-Provider Roaming in a Mixed-Mode Network](image)

8.3 Man-in-the-Middle Attack on UMTS

In this section, we present a man-in-the-middle attack on UMTS networks that results from a weakness in the inter-system roaming procedure between UMTS and GSM [122]. The attack allows an intruder to impersonate a valid GSM base station to a UMTS subscriber, regardless of the fact that UMTS authentication and key agreement are used. As a result, an intruder can eavesdrop on all mobile-initiated traffic.

Since the UMTS standard requires mutual authentication between MD and the network, so far UMTS networks have been considered secure against man-in-the-middle attacks. As already pointed out in Section 7.3.2.2, the network authentication defined in the UMTS standard depends on both the validity of the authentication token and the integrity protection of the subsequent security mode command.

As will be shown in the following section, both of these mechanisms are necessary in order to prevent a man-in-the-middle attack. Consequently, an attacker can mount
the impersonation attack since GSM base stations until the present time do not support integrity protection. Possible victims of this attack are all MDs that support the UTRAN and the GSM air interface simultaneously.

8.3.1 Protection of UMTS Subscribers with UMTS-Only User Equipment against Man-in-the-Middle Attacks in Standard UMTS Networks

In order to mount a man-in-the-middle attack in standard UMTS networks (see Section 7.3.2.2), an attacker would have to impersonate a valid network to the user. However, in the standard scenario, the combination of two specific security mechanisms protects MD from this attack: the authentication token AUTN and the integrity protection of the security mode command message (see Figure 7.4). The authentication token ensures the timeliness and origin of the authentication challenge and as such protects against replay of authentication data. The integrity protection prevents an attacker from simply relaying correct authentication information while fooling the respective parties into not using encryption for subsequent communication.

In particular, AUTN contains a sequence number SQN and a message authentication code MAC (see Equation (7.1)). On receipt of AUTN (see Figure 7.4), MD first extracts the sequence number SQN. If the sequence number is in the right range (see [9] for the details on the range), MD is assured that AUTN was issued recently by its HN. Otherwise, MD knows that either AUTN is the replay of an old value or the synchronization of the sequence number failed (a more detailed description of the procedure is given in [9]). MD then checks the message authentication code MAC. A correct MAC indicates that the authentication token was originally generated by HN. It is important to note that the correctness of the MAC and SQN being in range alone do not provide assurance to the mobile unit that the token was in fact received directly from the authorized network and not relayed by an attacker.

It is only the combination with an additional integrity protection of the control messages that prevents network impersonation. The security mode command message is not only integrity-protected but more importantly includes the security mode capabilities that the MD originally announced to the network on radio connection establishment. By checking the correctness of the integrity protection, MD is assured that this message was generated by a network entity in possession of the right integrity key. Furthermore, including the security capabilities of MD in the integrity-protected security mode command message is crucial in that it prevents both the mobile unit as well as the network from being fooled into using no encryption (or weak encryption) by an attacker. In order to succeed, an attacker would have to forge the integrity protection on the security mode command message, which is assumed to be impossible [5]. In case the security capabilities of the MD would not be repeated back in the security mode command message, the attacker could forge the protection, by replacing the original (not integrity-protected) security capabilities with its own and making the valid network integrity-protect the security mode command with these replaced capabilities. An attacker could therefore claim on behalf of the victim MD to support only the mandatory
encryption algorithms (instead of its original security capabilities).\footnote{Currently, only no encryption (UEA0) and UEA1 and one integrity-protection mechanisms UIA1 are defined, and all three are mandatory. Thus the security capabilities are currently always the same and could not be bid down even if no integrity protection was used.} In turn, the attacker would inform MD of the choice of no or weak encryption by the network in the security mode command.

\subsection*{8.3.2 Vulnerability of UMTS Subscribers Using a Combined UMTS/GSM Mobile Equipment to Man-in-the-Middle Attacks in UMTS Networks}

If a UMTS subscriber roams to a GSM BTS controlled by a 3G MSC, such as described in Case 4 (see Section 8.1.2), the cipher mode command message is neither integrity-protected, nor does it repeat the security capabilities previously announced by MD on radio connection establishment. Consequently, the message can be easily forged by an attacker. This limitation is due to the fact that GSM does not currently support integrity protection. In the following section, we will detail a man-in-the-middle attack which exploits this shortcoming. We show that an attacker can impersonate a GSM base station to a UMTS subscriber using the UMTS authentication procedure of the hybrid GSM/UMTS scenario.

In order to mount our attack, we assume that the attacker knows the IMSI of his victim, a reasonable assumption, as the attacker can easily get hold of the IMSI by initiating an authentication procedure prior to the attack (see Figure 7.4) and disconnecting from MD after receiving the IMSI. Note that by doing so, the attacker also learns the security capabilities of MD.\footnote{The feasibility of this attack is already recognized in the UMTS specification [7].} A MD that supports both the GSM and the UMTS radio interface, connects to UMTS whenever possible and connects to a GSM BTS only if no NodeB can be received with sufficient signal strength. An attacker can force a victim MD to connect to a BTS operated by himself instead of a UMTS NodeB operated by a legitimate network provider, e.g., by jamming the UMTS frequencies and sending its beacons with higher transmitting power than any of the present valid GSM BTSs. Our attack works in two phases:

\textbf{Phase 1:}

The attacker acts on behalf of the victim MD in order to obtain a valid authentication token AUTN from any real network by executing the following protocol:

1. During the connection setup, the attacker sends the security capabilities of the victim MD to the visited network.

2. The attacker sends the TMSI of the victim MD to the visited network. If the current TMSI is unknown to the attacker, he sends a fake TMSI (which eventually cannot be resolved by the network).
3. If the network cannot resolve the TMSI, it sends an identity request to the attacker and the attacker replies with the IMSI of the victim.

4. The visited network requests the authentication information for the victim MD from its HN.

5. HN provides the authentication information to the visited network.

6. The network sends $RAND_v$, AUTN to the attacker.

7. The attacker disconnects from the visited network.

Since none of the messages sent in steps 1 through 7 are protected by any means, the network cannot recognize the presence of the attacker. Consequently, the attacker obtains an authentication token which he in turn can use in Phase 2 of the attack to impersonate a network to the victim MD.\footnote{Note that Phase 1 does not cause a false location update for the victim MD, as location updates follow successful authentications only.}

**Phase 2:**

The attacker impersonates a valid GSM base station to the victim MD.

1. The victim MD and the attacker establish a connection and MD sends its security capabilities to the attacker.

2. The victim MD sends its TMSI or IMSI to the attacker.
3. The attacker sends MD the authentication challenge $RAND_U$ and the authentication token AUTN he obtained from the real network in Phase 1 of the attack.\footnote{The MD accepts the authentication token if the token is fresh, in other words, if not too much time has elapsed between Phase 1 and Phase 2.}

4. The victim MD successfully verifies the authentication token.

5. The victim MD replies with the authentication response.

6. The attacker decides to use “no encryption” (or weak encryption, like a broken version of the GSM-encryption algorithms; see, for example, [28]).

7. The attacker sends MD the GSM cipher mode command including the chosen encryption algorithm.

The attack does not allow the intruder to impersonate MD to the network at the same time. In order to allow for a regular use of the connection by the victim unit, the attacker has to establish a regular connection to a real network to forward traffic it receives from MD. As a side effect, the attacker has to pay the cost for this connection.

**Feasibility of the Attack:**

An attacker trying to impersonate a valid network to a UMTS subscriber has to overcome two difficulties: he has to send (or forward) a valid authentication token to the victim MD, and he has to make sure that no encryption is used after the authentication.
In our attack, the intruder solves the first problem by impersonating the victim MD to a real network in order to obtain a valid authentication token. This section is possible since none of the respective messages are encrypted or integrity-protected.

Requesting no or weak encryption is more difficult, as the radio access network decides which encryption algorithm is used (in both the GSM and the UTRAN case). The decision strongly depends on the security capabilities of MD, which are sent to the network during connection setup (see Figures 8.4 and 7.4). In principle, both radio access networks (i.e., GSM and UTRAN) allow “no encryption.”

In UTRAN the security mode command message that informs MD which algorithm to use is integrity-protected. The integrity protection alone, however, does not protect against network impersonation. The attacker could still fool a valid network into integrity-protecting a “no encryption” message with the right key by sending him false information about the encryption capabilities of the victim MD. But as UEA1 is mandatory and currently only UEA0 and UEA1 are defined, a bidding-down is currently not possible even if the integrity protection was fakeable. Furthermore, in the integrity-protected security mode command message, the network sends the security capabilities it received back to MD. Unless the attacker can forge the integrity check, MD would thus detect a bidding-down attack.

In the GSM case, though, integrity protection is currently not supported. As a consequence, the corresponding cipher mode command message is not integrity-protected, thus allowing an attacker to easily forge this message and fool the victim MD into using either no encryption or a weak encryption algorithm. Eventually, the attacker is able to eavesdrop on all mobile-initiated communication.

Our attack only works as long as the time gap between Phase 1 and Phase 2 is small enough so that no other authentication between the victim MD and another network takes place. Otherwise, the sequence number within the authentication token might be out of range.

As stated before, our attack does not work against mobile equipment that is capable of the UTRAN interface only. Yet, in the transition phase from GSM to UMTS, most users are expected to use equipment that is capable of both the UTRAN radio interface and GSM.

The UMTS specification includes an optional and not further specified display of the current encryption and integrity-protection state [9]. If this is implemented in the victim’s MD, the victim may be able to suspect the attack, depending on the details provided to the user. However, if, for example, MD only displays encryption on/off and the attacker uses a broken algorithm like A5/2 for encryption [28], the subscriber will not be able to detect the attack. Moreover, if MD displays UMTS/GSM authentication only, the victim may be mislead about his current level of protection if MD indicates UMTS authentication.

8.3.3 Countermeasures

Avoiding the attack by not allowing roaming to GSM is not an option, primarily for economical reasons. Consequently, the authentication procedure has to be changed in order to protect against the man-in-the-middle attack described above. In [122] we suggested
8.4 Conclusion

The UMTS specification allows for a variety of combinations of UMTS and GSM user equipment, subscriber identity modules, and radio access networks. In order to protect UMTS subscribers from attacks known to GSM networks, the UMTS specification does not allow UMTS-capable equipment to carry out GSM authentication and key agreement unless the network is incapable of UMTS authentication and key agreement [9]. However, in this chapter we have detailed an attack that shows that, due to the inter-operation with GSM, the use of the currently specified UMTS roaming authentication and key-agreement procedures are not sufficient in order to protect UMTS subscribers from man-in-the-middle attacks. Implementing countermeasures to thwart the attack will require modification of the GSM or the UMTS standard.
Chapter 9

Handover within and between GSM and UMTS

The GSM and the UMTS standards specify handover procedures of MDs within their HN in the same way as handover procedures of roaming MDs. Handover procedures within FN are controlled by FN. Handover procedures between different providers are not specified.

Candidate services for handover are incoming and outgoing phone calls (circuit-switched services), as well as ongoing GPRS sessions (packet-switched services).

Aside from handover within UMTS, the UMTS standard specifies handover procedures between a GSM BSS (also called GERAN in the UMTS standard) and UTRAN within mixed-mode networks, as well as between different network operators.

All handover procedures specified for UMTS and GSM are network-initiated mobile assisted procedures as modeled in Figure 3.5. MD periodically sends measurement reports to the visited network. These reports include information about the best received surrounding BTSs and Node Bs in MD’s current location. The algorithm by which the necessity of a handover is determined is not specified in the standard.

The handover procedures within GSM and UMTS make use of a security-context transfer in which the data protection key(s) used before handover is (are) transferred to the new radio access network and then reused. On inter-system handover between UMTS and GSM, the data-protection keys used before handover are additionally converted to the right length for the respective destination technology.

In this chapter, we describe the security-context transfer for all intra-provider and inter-provider handover procedures between and within GSM, UMTS, and mixed-mode networks. The handover procedures themselves are described in [11, 60]. The security-context transfer is specified in [9, 62]. Here, we present a comprehensive overview on all of the different SCT cases. Furthermore, we discuss whether these procedures meet the newly defined requirements for SCT and analyze their security with help of the general threat analyzes presented in Chapter 4.

Moreover, we detail the impact of GSM vulnerabilities on the security of UMTS and GSM subscribers roaming in a heterogeneous GSM/UMTS network world. This part of
the thesis has been published in [123] and is joint work with S. Wetzel. In particular, we show that a single handover of a USIM-equipped MD to a GSM network that uses a broken version of the A5 algorithm is sufficient to break the encryption of all post-handover UMTS traffic. GSM subscribers profit from the higher protection level of a UMTS connection only if they do not roam and are not handed over to the GSM (part of a) network between two authentications. We also study the impact of a two-sided GSM man-in-the-middle attack and show that such an attacker can together with his victim be handed over from GSM to UMTS. To secure UMTS access networks against GSM vulnerabilities, we suggest in [123] to integrate an additional new authentication into each inter-system handover procedure. The found vulnerabilities and our countermeasures were discussed by 3GPP [3] and part of our countermeasures were included in [4].

Outline. In Section 9.1, we describe the security-context transfer on intra-provider handover in GSM. This is followed by a brief description of the SCT on intra-provider handover in UMTS in Section 9.2. We detail the security-context transfer of inter-system handover between GSM and UMTS in Section 9.3. The impact of GSM vulnerabilities on UMTS due to the inter-system handover is discussed in Section 9.4. We close this chapter with a Conclusion in Section 9.5.

9.1 Analysis of Intra-Provider Handover in GSM

For GSM, only hard handover procedures are specified. As a consequence of the hierarchical structure of a GSM backbone network, four handover types of intra-provider handover are distinguished: handover within the same cell (handover between different time-slots), handover between different cells controlled by the same BSC, handover between BTSs controlled by the same MSC, and finally handover between different MSCs. In an intra-cell handover, the encryption and integrity-protection endpoint does not change. From a security point of view, these handover are irrelevant and we can concentrate on the three other handover types. A detailed description of the different handover procedures is given in [60]. We concentrate here on the transfer of the GSM-encryption key and the negotiation of security mechanisms only.

In accordance with the notation introduced in Part II, we will speak of the source BTS as the BTS to which MD is connected before the handover and of the destination BTS as the BTS to which MD is connected afterwards. Similarly, we will refer to the source MSC and the destination MSC as the MSCs that control the source BTS or the destination BTS. Note that source and destination MSC can coincide (intra-MSC handover). Furthermore, we refer to the MSC to which MD was connected during the call or GPRS-session establishment as the anchor MSC of a handover. It is important to note that the anchor MSC stays the same on subsequent handover and that it is the anchor MSC that stays responsible for the call. Due to the way intra-provider roaming is implemented in GSM (Section 7.2), the anchor MSC may differ from the MSC that was engaged in the last authentication of MD.

---

1 GSM, currently, does not support inter-provider handover.
On handover, a simple security-context transfer is used to transfer the last negotiated encryption key $K_c$ to the destination BTS. If the destination BTS is controlled by the source BSC, BSC simply forwards $K_c$ to BTS. If the destination BSC differs from the source BSC but both are controlled by the same MSC, the MSC transfers $K_c$ to the destination BSC, which in turn forwards it to the destination BTS. If the destination MSC differs from the source MSC, the latter one first transfers $K_c$ to the destination MSC, which in turn sends it to the destination BTS via its BSC. The GSM security-context transfer does not use a key-derivation function $kd$ to separate the keys used before from the ones used after handover. Consequently, neither R*-2 nor R*-3 can be met. GSM does not support tracking of the lifetime of an encryption key. Consequently, considerations with respect to the lifetime of a key cannot be taken into account during handover (R*-4).

The standard specifies that if encryption is disabled before handover, it will stay disabled after handover [62, 60]. How the A5 algorithm used after handover is negotiated is not further specified in the standard. It is, however, noted that the applied A5 algorithm may change after handover and the choice of the allowed algorithms is left to the destination MSC. As in the intra-provider case the destination MSC and the source MSC are operated by the same provider, both can be assumed to have the same policies, such that R*-5 is obsolete (see Section 6.3). As GSM does not currently support integrity protection, the handover command message and the messages exchanged to negotiate the encryption mechanism to use after handover are not integrity-protected. Thus, neither R*-8 nor R*-6 can be met. The destination MSC does not obtain any information on the previously used cipher suites (R*-1 not met). How handover requests are protected is not specified in the standard. It therefore depends on the network operator in question whether R*-7 is met.

Subsequent intra-provider handover between different MSCs are controlled by the anchor MSC. The anchor MSC transfers $K_c$ to the $k$-th destination MSC on a $k$-th order intra-provider handover in GSM.

In the following we analyze the security of GSM-handover procedures with the help of the threat analysis presented in Chapter 4.

As in GSM no key-establishment process $ke$ and no key-derivation $kd$ are used, GSM is vulnerable to the attack modules BAM*-1 to BAM*-4. Moreover, as will be discussed in Section 9.4.1 the GSM encryption algorithm A5/2 is broken, such that GSM is vulnerable to BAM*-5 if A5/2 is used. Whether or not GSM is vulnerable to BAM*-6 is unclear. In some cases GSM BTSs are connected via a radio link to the rest of the backbone network. In this case, an attacker can intercept the transfer of the encryption key $K_c$ from MSC to BTS and thus achieve BAM*-6(a). The same holds for BAM*-7(a). The other alternatives in BAM*-6 and BAM*-7 cannot easily be assessed without detailed knowledge of the implementations of network components.

As a consequence of its vulnerability to the above basic attack modules, GSM is vulnerable to the attack modules AM*-1 if A5/2 is used between MD and the $j$-th destination BTS and to AM*-4 if A5/2 is used between MD and the $k$-th destination BTS. If a GSM network is vulnerable to BAM*-6 (BAM*-7) it is also vulnerable to AM*-2 (AM*-5). As it is unclear whether GSM is vulnerable to BAM*-6 and BAM*-7 it is unclear whether GSM
Table 9.1: Candidates for New Attacks against GSM
standard, they could be implemented in the same way as intra-provider handover.

## 9.2 Intra-Provider Handover within UTRAN

Similar to GSM, UMTS supports intra-provider handover procedures within the same cell, within the same RNC, between RNCs controlled by the same MSC and between different MSCs. UMTS also supports subsequent intra-provider handover between different MSCs. These are controlled by the anchor MSC. As opposed to GSM, UMTS supports hard and soft handover procedures. A detailed description of the UMTS handover procedures can be found in [11]. We concentrate here on the security-context transfer during handover. Unless the RNC changes, no security context has to be transferred. We therefore concentrate on handover between different RNCs.

UMTS uses a simple security-context transfer without a key-derivation function during handover. The anchor MSC transfers the encryption and integrity-protection keys $CK$ and $IK$ to the destination MSC, which then forwards them to the destination RNC. Consequently, requirements R*-2 and R*-3 are not met by the security-context transfer used within UMTS. However, in UMTS MD keeps track of how long a cipher key and integrity-protection key pair has been used by means of counters that measure the amount of data thus far protected with these keys. The visited network can query MD about these counters. The counters are stored on the USIM card, such that it can be assumed that a subscriber cannot easily manipulate them. Like all other control traffic responses to queries of the current counter values are required to be integrity-protected on the air interface, such that they cannot be manipulated. Consequently, lifetime restrictions on data-protection keys can be enforced (R*-4).

As in GSM, encryption stays disabled after handover if it was disabled before handover. How the UEA and UIA algorithms for use after handover are negotiated is not further specified in the standard. In particular, it is unclear whether policies of HN can be enforced during the negotiation R*-5 and whether the negotiation is protected against bidding-down attacks R*-6. However, as integrity protection of control traffic is mandatory both before and after handover, handover command messages are always integrity-protected in UMTS (R*-8).

As in GSM, the destination MSC does not obtain any information on the previously used cipher suites (R*-1 not met) and how handover requests are protected is not specified in the standard. It therefore depends on the network operator in question whether R*-7 is met.

In the following we analyze the security of handover procedures within UTRAN with the help of the threat analysis presented in Chapter 4.

As in GSM, in UMTS no key-establishment process $ke$ and no key-derivation function $kd$ is used. Therefore, UMTS is vulnerable to the attack modules BAM*-1 to BAM*-4.

---

2Note that the anchor MSC and the destination MSC may coincide.

3This may be due to the fact that currently only one encryption algorithm and one integrity-protection algorithm is specified.
As opposed to GSM, to date no efficient attack against the UMTS encryption mechanism KASUMI is known, such that UMTS is currently secure against BAM*-5(a). Whether or not UMTS is secure against the other alternatives in BAM*-5, and whether or not UMTS is vulnerable to BAM*-6 and BAM*-7 is unclear and depends on the details of how the UMTS network components and their communication are protected. As in UMTS the encryption and integrity protection reaches back to the RNC, an attacker cannot intercept any key transfer even if NodeBs are connected to RNCs over a radio link.

As a consequence, UMTS is currently well-protected against the attack modules AM*-1, AM*-4, AM*-2, AM*-5, AM*-3, and AM*-6.

In UMTS an attacker cannot bid-down the security-mechanism negotiation as the security capabilities announced to the network by MD is returned to MD in an integrity-protected message (see Chapter 8). However, as in GSM, an attacker can watch out for network providers that command MDs to use “no encryption” in order to mount certain attacks.

As in UMTS control traffic is integrity-protected, it is currently impossible to fake handover commands in UMTS. Whether it is easy to fake handover requests in UMTS is hard to tell, as very little is publicly known as to how providers protect the communication on their backbones.

In summary, the attacks collected in Table 9.2 are candidates for being mountable in UMTS (see also Table 4.3).

Table 4.3 shows that if the UMTS network components and their communication are well-protected, UMTS is secure against most of the identified attacks. The only attacks that are left to be considered in this case are the attacks A*-4, A*-21, and A*-24 that make use of a disabled encryption between a mobile device and a previous source Node B. We will discuss the impact of two GSM vulnerabilities, namely supporting a breakable encryption mechanism and supporting fake base station attacks on inter-operating GSM/UMTS networks in the next section.

### 9.3 Inter-System Handover between GSM and UMTS

To facilitate the transition from GSM to UMTS, the UMTS standard allows for handover procedures between UMTS and GSM radio access networks. These may be operated by the same or by different network providers. As within GSM and UTRAN, a security-context transfer with key derivation is used to secure user and control traffic after handover. As opposed to the homogeneous GSM and UMTS cases, an inter-system handover makes conversions of GSM keys into UMTS keys necessary. Which keys are converted into which and who converts and transfers them on the network side depends on the smart-card type used, the capabilities of the anchor and destination MSC, and the capabilities of the destination radio access network.

In principal, on inter-system handover from a UMTS RNC as anchor to a GSM BSS as destination radio access network, the UMTS keys $IK$ and $CK$ used before handover are converted into a GSM key $KC$ using the conversion function $c_3$ (see Equation (8.3)). $KC$ is
### Table 9.2: Candidates for Attacks against UMTS

<table>
<thead>
<tr>
<th>Attack Module</th>
<th>Used in Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depends on protection of network components</td>
<td>A*-7, A*-9, A*-14, A*-15, A*-18, A*-20, A*-23, A*-25</td>
</tr>
<tr>
<td>Depends on protection of network components</td>
<td>A*-6, A*-10, A*-15, A*-16, A*-13, A*-17, A*-23, A*-25</td>
</tr>
<tr>
<td>Unclear</td>
<td>A*-12</td>
</tr>
<tr>
<td>Depends on protection of network components</td>
<td>A*-1</td>
</tr>
<tr>
<td>Depends on protection of network components</td>
<td>A*-2, A*-3, A*-20</td>
</tr>
<tr>
<td>Unclear</td>
<td>A*-5, A*-15, A*-20</td>
</tr>
<tr>
<td>Depends on protection of network components</td>
<td>A*-5, A*-9, A*-14, A*-20</td>
</tr>
<tr>
<td>Depends on protection of network components</td>
<td>A*-10, A*-17</td>
</tr>
<tr>
<td>Unclear</td>
<td>A*-8</td>
</tr>
<tr>
<td>Not possible because of bid-down protection</td>
<td>A*-8, A*-11, A*-19, A*-22</td>
</tr>
<tr>
<td>Possible if attack observes use of “no encryption”</td>
<td>A*-4, A*-21, A*-24</td>
</tr>
<tr>
<td>Not possible</td>
<td>A*-13</td>
</tr>
<tr>
<td>Unclear</td>
<td>A*-27</td>
</tr>
<tr>
<td>Impossible because handover command integrity-protected</td>
<td>A*-14, A*-15, A*-16, A*-26</td>
</tr>
</tbody>
</table>
then used to protect data and control traffic after handover. Upon handover from a 2G MSC as anchor to UTRAN, the GSM key $K_c$ is converted into the UMTS keys $IK$ and $CK$ by the conversion functions $c_4$ and $c_5$ (see Equations (8.1) and (8.2)). Which type of keys are transferred thus depends on the capabilities of the anchor and the destination network. Which keys are generated during authentication depends on the capabilities of the authenticating MSC, as well as the smart-card type.

By construction of the key-conversion functions, subsequent conversions from GSM to UMTS and back to GSM retrieve the original GSM key:

$$c_3(c_4(K_c), c_5(K_c)) = K_c$$

The key-conversion functions $c_3$, $c_4$, and $c_5$ are very similar to the ones suggested in Chapter 6.1 and exhibit the desired property with the same argument (see Section 6.1.1.3). However, no additional key-derivation function separates the keys used before from those used after handover, such that $R^*-2$ and $R^*-3$ are not met. In particular, each intermediate UMTS network obtains knowledge of previously and subsequently used data-protection keys. The requirements $R^*-1$ is likewise not met. Violations of key lifetime restrictions can be detected by any UMTS network MD is handed over to, such that $R^*-4$ is met in part.

The standard specifies that if encryption was disabled in the GSM BSS (UMTS RNC) before, it will stay disabled after handover to a UMTS RNC (GSM BSS). If encryption was enabled, the source radio access network sends MD’s security capabilities to the destination radio network, which in turn chooses one of the supported mechanisms. MD is informed of this choice in the handover command message. It is, however, unclear whether AN’s policies could be enforced during handover ($R^*-5$) and whether the negotiation between AN and DEST can be protected against bidding-down ($R^*-6$). Upon handover from UMTS to GSM, the handover command message is integrity-protected. However, upon handover from GSM to UMTS, this is not the case ($R^*-8$). The standard does not specify how messages exchanged between different network providers are protected. It therefore depends on the network providers in question whether the handover request message that includes the security context, is encrypted and integrity-protected ($R^*-7$).

In the following, we detail the key transfer for all possible constellations of anchor MSCs, authenticating MSCs, and destination MSCs, first for USIM-equipped MDs and then for SIM-equipped MDs.\(^4\)

### 9.3.1 Key Transfer on Handover of USIM-Equipped MDs

We start with the cases in which the anchor and the authenticating MSC coincide (see Figure 9.1).

#### Case 1: A USIM-equipped MD is authenticated by a 3G MSC. During authentication and key agreement, the UMTS keys $CK$ and $IK$ are established between the 3G MSC and MD. If MD is subsequently handed over from the authenticating MSC as anchor network to

\(^4\)Note that MD either has a SIM inserted or a USIM, but never both at once.
9.3 Inter-System Handover between GSM and UMTS

Case 1:
- USIM
- 3G MSC
- 3G MSC
- IK, CK

Case 2:
- USIM
- 3G MSC
- 2G MSC
- Kc = c3(CK,IK)

Case 3:
- USIM
- 2G MSC
- 3G MSC
- Kc = c3(CK,IK)

Case 4:
- USIM
- 2G MSC
- 2G MSC
- Kc = c3(CK,IK)

Case 5:
- USIM
- 3G MSC
- 2G MSC
- 3G MSC
- Kc = c3(CK,IK)
- Kc = c3(CK,IK)

Case 6:
- USIM
- 3G MSC
- 2G MSC
- 3G MSC
- 3G MSC
- Kc = c3(CK,IK)
- Kc = c3(CK,IK)
- CK*, IK*

Figure 9.1: USIM Handover Cases with Authenticating MSC = Anchor MSC

Figure 9.2: Additional USIM Handover Cases with Authenticating MSC ≠ Anchor MSC

As described in Section 8.2, the UMTS standard allows for several inter-system intra-provider roaming in mixed-mode GSM/UMTS networks. This is why the authenticating MSC may differ from the anchor MSC. Consequently, we have to consider two additional handover scenarios (see Figure 9.2):

Case 5: As in Case 1, MD is authenticated via a 3G MSC. While still in idle mode, MD
roams to a 2G MSC operated by the same provider. The 3G MSC transfers $K_c$ to the 2G MSC. Subsequently, MD places or receives a phone call or establishes a GPRS connection via the 2G MSC. Upon handover, the 2G MSC is now the anchor network. The 2G MSC transfers the GSM key $K_c$ to the 3G MSC. Case 5 is thus effectively the same as Case 3.

**Case 6:** As in Case 5, MD is authenticated via a 3G MSC. While still in idle mode, MD first roams to a 2G MSC and subsequently to a 3G MSC. Accordingly, the authenticating network derives the GSM key from the UMTS keys established during authentication and transfers it to the 2G MSC. The 2G MSC in turn transfers it to the 3G MSC. Subsequently, MD places or receives a phone call or establishes a GPRS connection via the 3G MSC. Upon handover, the 3G MSC is then the anchor network. The important change in this situation is that the 3G MSC on handover to a 3G MSC as destination now transfers the derived UMTS keys $C_{K^*} = c_4(K_c)$ and $I_{K^*} = c_5(K_c)$, rather than the UMTS keys originally established during the authentication.

It is important to note that in case the anchor network is a 3G MSC but MD is connected to it via a GSM BSS, the 3G MSC nevertheless stores the originally received UMTS keys for MD. Consequently, the key transfer, e.g., in Case 1, does not depend on whether MD is connected to the 3G MSC via a GSM BSS or UTRAN.

### 9.3.2 Key Transfer on Handover of SIM-Equipped MDs

For SIM-equipped MDs, the situation is simpler. Regardless of the type of the authenticating MSC, the authentication is always based on GSM-authentication vectors. The GSM key is then, if necessary, converted into the *pseudo-UMTS keys* $C_{K^*} = c_4(K_c)$ and $I_{K^*} = c_5(K_c)$. If these pseudo-UMTS keys are reconvened into a GSM key with the help of $c_3$, the originally established $K_c$ is recovered. This prevents a subsequent downgrade of $K_c$. Figure 9.3 illustrates the different handover situations depending on the type of the anchor MSC. As opposed to the USIM case, we do not have to take the authenticating MSC into account.

In the first two cases, the anchor MSC is a 3G MSC. If the destination MSC is a 3G MSC as well, the anchor MSC transfers the pseudo-UMTS keys $C_{K^*}$ and $I_{K^*}$ (Case 1). If
the destination MSC is a 2G MSC, the anchor MSC first recovers the original $K_c$ from $CK^*$ and $IK^*$ and then transfers $K_c$ (Case 2).

In the two latter cases, the anchor MSC is a 2G MSC. The anchor MSC transfers the original GSM key $K_c$ to the 2G or 3G destination MSC (Case 3, Case 4).

### 9.4 Impact of GSM Vulnerabilities on UMTS via Handover

GSM suffers from various security weaknesses. Recently, Barkan, Biham, and Keller presented a ciphertext-only attack on the GSM-encryption algorithm A5/2 which recovers the encryption key from a few dozen milliseconds of encrypted traffic within less than a second [28]. Furthermore, it is possible to mount a man-in-the-middle attack in GSM during authentication, which allows an attacker to make a victim MD authenticate itself to a fake base station, which in turn forwards all authentication traffic to a real network, thus impersonating the victim MD to a real network and vice-versa.

In this section, we analyze the impact of GSM-encryption attacks that recover the encryption key and the impact of the man-in-the-middle attack on the security of networks that employ UMTS and GSM base stations simultaneously [123].

We suggest protecting UMTS connections from GSM attacks by integrating an additional authentication and key agreement on inter-system handover between GSM and UMTS. This countermeasure was discussed by 3GPP in [3].

#### 9.4.1 GSM Attacks

##### 9.4.1.1 Attacks on GSM Encryption

The GSM-encryption algorithms A5/1 and A5/2 were originally kept secret. However, in 1994, a sketch of the design of A5/1 was leaked. In 1999, Briceno, Goldberg, and Wagner [41] reverse-engineered the exact design of both algorithms. Since then, various attacks on the algorithms were published. The first publicly available cryptanalysis of A5/1 was published by Golic in 1997 [77]. Other attacks soon followed in [34], [33] and [57]. For A5/2, Goldberg et al. [76] first devised a known plaintext attack, which requires the attacker to know the XOR of two plaintexts that are exactly $2^{11}$ frames apart. Subsequently, Petrovic et al. [143] proposed an attack which allows predicting the key stream produced by A5/2 from the knowledge of a few hundred known ciphertext/plaintext bit pairs. The strongest attack on A5/2 known to date was described by Barkan, Biham, and Keller in [28]. Their ciphertext-only attack requires only a few milliseconds of encrypted voice traffic (4 frames) to be passively intercepted by the attacker in order to allow the recovery of the corresponding encryption key $K_c$ within less than a second. The attack works because encryption is applied after error-correction. This leads to known linear relationships between the plaintext bits to be encrypted. The authors also describe three active attacks that use the A5/2 attack to break the encryption if A5/1 or A5/3 are used.

In the following section, we concentrate on the type of encryption attacks (such as the A5/2 attack of Barkan, Biham, and Keller) that recover the encryption key $K_c$. The
impact of encryption attacks that merely predict the key stream output of a GSM-encryption algorithm is not studied here.

9.4.1.2 Man-in-the-Middle Attack(s) on GSM

GSM is vulnerable to a man-in-the-middle attack, which allows an attacker to impersonate a fake base station to a victim MD [67]. In order to mount this attack, the attacker forces the MD to connect to a fake base station by broadcasting beacons. If the MD is in stand-by mode, it will always connect to the base station it receives best. Thus, the attacker can make MD connect to him by sending its beacons with a higher transmission power than any real base station. By requesting to turn off encryption in the GSM cipher mode command he sends to MD, the attacker can disable the encryption between MD and the fake base station. As a result, an intruder can eavesdrop on all mobile-initiated traffic. Unless the attacker cannot impersonate MD to a real network as well, MD will be unreachable for incoming traffic.

An attacker can also easily impersonate a victim MD to a real network during authentication by simply forwarding the authentication traffic between them. Disabling the encryption between itself and the real network is, however, not easy. The GSM standard mandates a MD to support A5/0, A5/1 and A5/2 [62, 63]. An attacker cannot thus manipulate the security capabilities of a MD to include A5/0 only. Nevertheless, if an attacker knows that a certain legitimate network always disables encryption, he can succeed with a two sided man-in-the-middle attack. It is interesting to note that in this attack scenario, an attacker actually makes a victim MD use a network that does not provide encryption rather than waiting for MD to connect to this network itself.

9.4.2 Impact of Encryption Attacks

In this section, we discuss how an attack that recovers the GSM encryption key (like the A5/2 attack described above\textsuperscript{5}) influences the network security in areas where both GSM and UMTS radio access technologies are available simultaneously and inter-operate by roaming and handover. As discussed earlier, these kind of areas already exist and will continue to exist until the last GSM subscriber has updated his subscription to a UMTS subscription and the last base station that is capable of GSM only has been replaced. Our discussion is independent of whether the radio access networks are operated by the same or different network operators.

For our analysis, we reuse the cases distinguished in Section 9.3. Note that handover between the same types of radio access networks do not have to be considered here, as a successful GSM key-recovery attack does not have an additional impact through them. We start off with handover of a USIM-equipped MD and then study the impact on the security of SIM-equipped MDs. Note that the numbering in this section is different from the one used in [123].

\textsuperscript{5}Note that other attacks that recover the GSM-encryption key have the same impact.
9.4.2.1 Impact on USIM-Equipped MDs

We reuse Figure 9.1 and the cases distinguished in Section 9.3 and start with the case that the authenticating MSC is the anchor MSC of the handover.

Case 1: A USIM-equipped MD is authenticated by a 3G MSC and is handed over by this MSC as anchor to a GSM base station that is connected to a 3G MSC.

During authentication, the UMTS keys $IK$ and $CK$ were generated on the USIM and in HN. These keys were used before handover. Upon handover to a GSM base station, the keys are converted into a GSM key $K_c$ by means of $c_3(IK, CK) = CK_1 \oplus CK_2 \oplus IK_1 \oplus IK_2$ in both MD and the source MSC. The GSM key and the UMTS keys are transferred from source 3G MSC to the destination 3G MSC. The 3G MSC stores the UMTS keys for subsequent handover back to UMTS. The GSM-encryption key $K_c$ is used to encrypt the communication between MD and the GSM base station after handover.

If an attacker can break the encryption algorithm used after handover and he recovers the encryption key $K_c$, then this knowledge of $K_c$ leaks 64 bits of information of the 256-bit UMTS keys used before handover.

Case 2: A USIM-equipped MD is authenticated via a 3G MSC and is handed over by this MSC as anchor to a 2G MSC.

During authentication, the UMTS keys were generated on the USIM card and in HN. Before handover, the UMTS keys were used to secure the communication between MD and the source MSC. Upon handover to GSM, the keys are converted in MD and in the source MSC to a GSM key $K_c$ using the conversion function $c_3$. $K_c$ is then transferred from the source 3G MSC to the destination 2G MSC. Unlike in Case 1, the UMTS keys are not transferred because the 2G MSC cannot handle them.

Breaking the GSM encryption after handover reveals 64 bits of the UMTS key material used before handover to the attacker.

Case 3: A USIM-equipped MD is authenticated by a 2G MSC and is then handed over by this 2G MSC as anchor to UTRAN.

Since the 2G MSC is not able to perform a UMTS authentication, MD and HN have to derive a GSM-authentication vector from a UMTS-authentication vector. Consequently, during the authentication, the UMTS keys $IK$ and $CK$ were generated on the USIM card as well as in the home network and immediately converted into the GSM key $K_c$ using $c_3$. Upon handover to UMTS, the GSM key $K_c$ is converted into the pseudo-UMTS keys $IK^* = c_5(K_c)$ and $CK^* = c_4(K_c)$. These keys are different from the keys $IK$ and $CK$ that were generated on the USIM card during authentication.

If an attacker can recover the GSM-encryption key $K_c$ before handover, he can use $K_c$ to compute the UMTS keys $IK^*$ and $CK^*$. Thus, the encryption and integrity protection after handover are broken. Moreover, handover reveals 64 bits of information of the UMTS keys $IK$ and $CK$ generated during authentication. Note that in this case, the encryption is broken for each subsequent handover, as all subsequent handover are controlled by the 2G anchor network.

Case 4: A USIM-equipped MD is authenticated by a 3G MSC. While connected to a GSM
base station controlled by the authenticating 3G MSC, MD switches from idle to active mode. It is then handed over to a UMTS base station controlled by the authenticating (= anchor MSC) or another 3G MSC.

Since the 3G MSC can carry out a UMTS authentication, MD and HN generate the UMTS-authentication vector and HN transfers this vector to the authenticating MSC. The GSM base station transparently forwards the authentication traffic. During the authentication, the UMTS keys were generated on the USIM card and in HN. After a successful authentication, the UMTS keys are converted into a GSM-encryption key $K_c$ using the conversion function $c_3$. Instead of converting the GSM key into a UMTS key upon handover to UMTS, the original UMTS keys stored in the anchor 3G MSC are transferred from the anchor MSC to the destination MSC and from the destination MSC to the UMTS NodeB.

An attacker who can recover the GSM-encryption key $K_c$ learns 64 bits of information on the UMTS key material used after handover.

To describe the additional cases arising from different anchor and authenticating MSCs we refer to Figure 9.2.

Case 5: A USIM-equipped MD is authenticated by a 3G MSC, subsequently roams to a 2G MSC, and is then handed over from the 2G MSC as anchor to UTRAN.

As in Case 1, MD is authenticated by a 3G MSC with the help of a UMTS-authentication vector obtained from HN. On intra-provider roaming to the 2G MSC, the authenticating 3G MSC converts the UMTS keys into the GSM key $K_c$ by $c_3$. On handover from the 2G MSC as anchor to a 3G MSC, the anchor MSC transfers the GSM key $K_c$. The destination 3G MSC converts $K_c$ into the pseudo-UMTS keys $CK^*$ and $IK^*$. If an attacker can recover the GSM-encryption key $K_c$ while MD is connected to the anchor 2G MSC, he cannot only recover 64 bits of the information on the UMTS keys originally generated, but he can completely break the post-handover communication protected by $IK^*$ and $CK^*$.

Case 6: A USIM-equipped MD is authenticated by a 3G MSC and first roams to a 2G MSC and then subsequently roams to a 3G MSC. With the 3G MSC as anchor, it is then handed over to another UTRAN.

In this case, the UMTS keys are generated during authentication. Upon roaming to the 2G MSC, only the derived GSM key is transferred. The 2G MSC subsequently transfers the $K_c$ to the 3G MSC. The 3G anchor MSC then transfers the pseudo-UMTS keys to the destination MSC. If an attacker gets hold of $K_c$ while MD is connected to the 2G MSC, all subsequent communication between MD and a network are broken, as from this point on, MD and the network either use the broken GSM key or use the pseudo-UMTS keys that can be easily recovered from $K_c$ by $c_4$ and $c_5$.

In summary, a single handover or intra-provider roaming to a GSM base station connected to a 2G MSC breaks all post-handover and post-roaming communication of a USIM-equipped MD. If a USIM-equipped MD is active while connected to a GSM base station that is controlled by a 3G MSC, this reveals 64 bits of the UMTS key material used before and after handover.
9.4.2.2 Impact on SIM-Equipped MDs

Case 1: A SIM-equipped MD is authenticated by a 2G MSC and is handed over to UMTS.

During GSM authentication, the encryption key $K_c$ was generated on the SIM card as well as in HN. It was used to protect the communication between MD and the GSM base station. Upon handover to UMTS, MD and the destination MSC convert $K_c$ into the pseudo-UMTS keys $IK^*$ and $CK^*$ using the conversion functions $c_4$ and $c_5$. If the MSC before handover is a 3G MSC, it is this MSC that converts the GSM key and sends it to the destination MSC. If the source MSC is a 2G MSC, the GSM key is sent to the destination MSC, which then converts the GSM key into UMTS keys.

If an attacker can break the GSM-encryption algorithm used before handover, then the attacker knows the encryption key $K_c$. He can then also convert $K_c$ into the UMTS keys using $c_4$ and $c_5$ and thus break the UMTS encryption and integrity protection after handover. The attacker can then eavesdrop on the communication between MD and the base station and can insert and manipulate traffic between them.

Case 2: A SIM-equipped MD is authenticated by a 3G MSC and connected to UTRAN when handed over to GSM.

During authentication, the GSM-encryption key $K_c$ was generated on the SIM card and in HN. The MSC of the visited network and MD both converted $K_c$ into the UMTS keys $IK = c_5(K_c)$ and $CK = c_4(K_c)$. The UMTS keys were used to encrypt and integrity-protect the communication between MD and the UMTS base station before handover. Upon handover to GSM, the original encryption key $K_c$ is recovered in MD and in the source MSC by means of $c_3(CK, IK) = c_3(c_4(K_c), c_5(K_c)) = K_c$ and sent to the GSM base station.

If an attacker can break the GSM-encryption algorithm used after handover and he can recover the GSM-encryption key $K_c$, he can also compute the UMTS keys $IK$ and $CK$ used before handover using the conversion functions $c_4$ and $c_5$. If the attacker has recorded the communication between the UMTS base station and MD before handover, he can now decrypt the recorded traffic.

In summary, a SIM-equipped MD benefits from the higher UMTS security level only if it is not handed over to a GSM base station during an ongoing connection. Similarly, it profits only if it does not roam to the GSM part of the network provider, by which it was authenticated.

9.4.2.3 Impact of a Two-sided Man-in-the-Middle Attack

A man-in-the-middle attack, as described in Section 9.4.1.2, can occur on any GSM authentication. As GSM subscribers and UMTS subscribers equipped with a suitable MD can connect to a GSM base station and be authenticated in GSM style, both types of subscribers are vulnerable to this attack.

Assume that a subscriber has caught a two-sided man-in-the-middle attacker and the attacker and MD move out of range of the serving GSM base station, to which the attacker originally impersonated the victim. As described earlier, the attacker has disabled the encryption between himself and the network as part of the attack. Consequently, upon
handover to UMTS, the encryption is not enabled because it was disabled before handover. However, in order for the man-in-the-middle attack to carry over to UMTS, the attacker has to master the integrity protection of the control messages between MD and the UMTS base station, which is begun right after handover.

In order for the attacker to continue to impersonate the victim MD to the network, the attacker has to send correct integrity-protected messages to the network. The attacker cannot generate these messages himself, but he can force MD to generate them instead: the attacker simulates a handover to a UMTS base station to MD by impersonating the GSM base station and the UMTS base station at the same time. The attacker sends a handover command to MD that tells MD to connect to the fake UMTS base station. Depending on whether the last authentication was a GSM authentication or a UMTS authentication, the subscriber either converts the GSM key into UMTS keys or activates the stored UMTS keys for use after handover. Since MD will integrity-protect the messages, the attacker only needs to transparently forward these messages to the real UMTS base station.

Note that the impact of this attack differs from the impact of the encryption attacks. A man-in-the-middle attack does not depend on any type of broken encryption algorithm and thus always has an impact on inter-operating UMTS/GSM networks.

9.4.3 Countermeasures

In order to protect from the A5/2 attack in general, the 3GPP currently discusses disabling A5/2. While this action would protect users from the concrete threat of the attack to GSM networks and from the impact of the attack on inter-operating UMTS/GSM networks, the threat of similar attacks recovering the encryption key on inter-operating UMTS/GSM networks remains. Furthermore, disabling A5/2 does not protect from carrying over man-in-the-middle attacks from GSM to UMTS.

In [123] we proposed integrating an additional UMTS-authentication and key-agreement procedure in connection with inter-system handover in order to secure the UMTS part of the network against GSM-encryption attacks. Newly generated UMTS keys have no known relation to a broken GSM key and a newly generated GSM key does not reveal any information about formerly used UMTS keys.

Upon handover from UMTS to GSM, the new authentication would have to be carried out while the subscriber is still connected to a UMTS base station. In other words, a new authentication is performed whenever a subscriber enters a UMTS cell that is a border cell to a GSM part of the network. If the newly generated keys are UMTS keys, MD and the 3G MSC convert them into a GSM key $K_c$ using $c_3$. The MD and the serving 3G MSC store this key until the actual handover to GSM takes place. Upon handover to GSM, the source MSC transfers the key to the destination MSC, which in turn forwards it to the base station.

In the case of a handover from GSM to UMTS where the GSM base station is connected to a 3G MSC, the new authentication can be carried out before the actual handover. In this case, the MSC can initiate an authentication as soon as MD enters the cell. Since the 3G MSC can do a UMTS authentication, it can authenticate the MD using the respective
procedure as described in Section 8.1. Upon handover, the newly established keys are sent to the UMTS base station.

In case of a handover from GSM to UMTS where the GSM base station is connected to a 2G MSC, the 2G MSC is not capable of performing a UMTS authentication. Authenticating before handover, therefore does not protect from a man-in-the-middle attack. Instead, the new authentication must take place right after the handover. While this implies that an attacker can eavesdrop on the first few seconds of the connection to the UMTS network, the attacker will be shut out as soon as the authentication is successfully completed.

In order to avoid unnecessary authentications, sophisticated methods can be used to determine whether a MD that is currently located in a GSM cell is really going to be handed over to UMTS [103].

The above countermeasure was discussed by 3GPP in [3]. It was decided to adopt our countermeasure in case of intra-provider roaming from GSM to UMTS, i.e., to recommend the use of a new authentication in UMTS to all providers [4]. If these recommendations are followed, a USIM equipped MD in idle mode will be newly authenticated when roaming to UTRAN. Although the UMTS standard theoretically allows for authentication and key agreement during an ongoing connection (this would be necessary to newly authenticate MD on handover to UTRAN) there is some doubt in 3GPP whether any manufacturers will actually implement this feature. Rather than adopting our countermeasure on handover to UTRAN, 3GPP recommends in [4] ensuring that all 2G MSCs that support handover to 3G MSCs are capable of authentication based on UMTS-authentication vectors. As the key transfer in UMTS is AN-controlled, this countermeasure guarantees that for USIM-equipped MDs, UMTS keys are transferred on each handover to UTRAN.

Our original suggestion, an additional UMTS authentication during an ongoing connection immediately after handover, also prevents man-in-the-middle attackers to be carried over from GSM to UMTS. The countermeasures currently discussed in 3GPP do not protect against this second type of attack, because the authentication between a UMTS base station and a UMTS subscriber is secure against man-in-the-middle attacks, while an authentication based on UMTS-authentication vectors but carried out via a GSM base station is not (see Section 8.3). Nevertheless, this problem will be resolved as soon as the GSM integrity-protection enhancement [2] is implemented and made mandatory for all types of MDs.

9.5 Conclusion

In this chapter, we have described the security-context transfer during the various types of inter- and intra-system handover procedures within and between GSM and UMTS networks. We discussed to what extent they meet the security requirements defined in Chapter 4 and Chapter 6 and found that most of these requirements are not met. As a consequence, inter-system handover between GSM and UMTS allow GSM vulnerabilities to have an impact on UMTS networks.

In particular, we have described the impact of GSM encryption and man-in-the-middle attacks on the security of inter-operating UMTS/GSM networks. We have shown that for
GSM subscribers, a single handover to GSM breaks all pre-handover and post-handover UMTS communication. For UMTS subscribers, a handover to a GSM base station that is connected to a 3G MSC reveals 64 bits of information on the key material used in pre-handover or post-handover UMTS communication. The impact of a handover to a 2G MSC is even worse, as a single handover to a GSM base station breaks the encryption and integrity protection of all post-handover UMTS communication completely.

Furthermore, we have discussed that handover procedures allow man-in-the-middle attackers to be transferred from GSM to UMTS.

In order to thwart the attacks, we have proposed carrying out an additional authentication and key agreement in connection with inter-system handover.
Part IV

Inter-Provider Handover and Roaming in WLANs
PART IV IN THE GENERAL CONTEXT

In this part, we detail our new roaming and handover approaches in the context of WLANs. We describe the WLAN system and security architecture in Chapter 10. In Chapter 11, we introduce a new roaming authentication protocol EAP-TLS-KS that implements the secret-sharing approach introduced in Chapter 2 of Part I. In Chapter 12, we detail the history-enriched policy-based approach introduced in Chapter 5 of Part II for inter-provider handover between WLANs.
Chapter 10

System Model and WLAN Security

Wireless local area networks have become more and more widespread. Universities offer wireless Internet access to students and staff, companies allow their employees to access their Intranet wirelessly, hotspot providers offer wireless Internet access in public areas like airports and coffee shops, and even many technically non-interested people connect their home-computing environment wirelessly.

The most widespread wireless LAN standard is currently the 802.11 standard of the IEEE. Early versions of this standard offer data rates of 1-2 Mbit/s. Current versions allow for data rates of up to 54 Mbit/s. Besides the relatively low bandwidth of early versions, they have another major drawback: the original security architecture, Wired Equivalent Privacy (WEP), designed for the standard, was completely broken by the end of 2001 (see Section 10.2).

The complete break of the WEP security architecture led to many different proprietary security solutions for WLANs. The resulting incompatibilities among equipment of different operators called for a new standardized solution. In July 2004, the IEEE finally adopted a new standard: 802.11i. In the meantime, the Wireless Fidelity Alliance\footnote{The Wireless Fidelity Alliance (Wi-Fi Alliance) is a non-profit organization that tests the interoperability of WLAN products and certifies them with the Wi-Fi certificate.} summarized part of the standard that was agreed upon already in late 2002 as Wi-Fi-Protected Access (WPA)[180]. The main difference between WPA and 802.11i (sometimes also referred to as WPA2) is that WPA supports only the stream cipher RC4 (see e.g. [66]), while 802.11i additionally supports the Advanced Encryption Standard (AES) [48].

Outline. In this chapter, we first describe the system model of a WLAN in Section 10.1. We then briefly discuss the original WEP security architecture and its shortcomings in Section 10.2. In Section 10.3, we give an overview of 802.11i.
10.1 System Model

The 802.11 standard [91] specifies two modes of operation for WLANs, the *ad hoc* mode and the *infrastructure* mode. Here, we only describe the latter one. The network access points in an infrastructure mode WLAN are referred to as Access Points (APs). The standard specifies the physical and the MAC layer of the air interface between a MD, typically a laptop or a PDA, and an AP. Several APs are interconnected via a Distribution System (DS). The standard only specifies the required services of a DS, not the DS itself. However, in practice, primarily Ethernet (IEEE 802.3) is used to interconnect WLAN APs [70]. An access router typically connects the DS to the outside world, e.g. the internet. In case of the new security architecture, the standard also specifies an additional entity for authentication purposes, namely an authentication server. Figure 10.1 illustrates a typical wireless LAN network architecture.

10.2 WEP

The WEP security architecture aimed to provide privacy equivalent to a local area network by replacing the physical protection of the sockets with a challenge response authentication and the missing physical protection of a wire with encryption on the air interface. Protecting legitimate users against each other was not part of the original security goals.

As a consequence, WEP uses one single 64- or 128-bit secret key, also referred to as WEP-key, that is shared between all pre-registered MDs and all APs as sole credential. This one key is used for both authentication and encryption of the air interface between each MD and the AP with which it is currently associated. Legitimate users can thus by design easily impersonate each other and decrypt each other’s traffic intercepted on the air interface.

WEP supports two authentication methods, *Open Authentication* and the *Shared-Key Authentication*. The authentication endpoint on the network side is thus the network access point (NAP = AS). In Open Authentication, any MD can join the network without any
prior authentication. In Shared-Key Authentication, MD has to prove the possession of the shared secret key by a simple challenge-response mechanism: AP sends a random challenge to MD and MD answers with a RC4-encrypted response. This mechanism has been shown to be weak in [20] and [39]. As RC4 is a stream cipher, a challenge-response pair reveals the key stream used to encrypt the response. An attacker can thus use an intercepted challenge-response pair to recover a valid key stream and subsequently successfully authenticate itself to any access point of the WLAN.\(^2\)

WEP supports only one encryption mechanism, the RC4 stream cipher. RC4 is used with the shared secret key and a per-packet initialization vector as input. The initialization vector is chosen by the sender and transmitted to the receiver in the clear. This particular mode of RC4 has been broken. The attack, described in [66], recovers the secret key from encrypted traffic intercepted on the air interface between one or more MDs and an access point within a few seconds. The attack was successfully implemented (see [165]) such that nowadays the WEP key shared between all users and all access points of a WEP-protected WLAN can be recovered by an attacker with a laptop and open source software.

For integrity protection in WEP, a Cyclic Redundancy Check (CRC-32) is appended to a packet before it is encrypted. This integrity protection has been shown to be ineffective in [39]. The linearity of the CRC-32 combined with the linearity of the stream cipher makes it easy to change messages and append a valid CRC-32.

Note that in WEP, the authentication server, as well as the encryption and integrity-protection endpoint, coincides with the network access point (NAP = EIPE = AS). Consequently, no key transfer from AS to EIPE is necessary. As only one pair of encryption and integrity-protection mechanisms and only one authentication protocol are specified, no security-mechanism negotiations are necessary. As the pre-shared secret key is used directly for authentication and encryption, no key agreement or key establishment is used.

It is important to note that intra-provider roaming and handover within a WEP-protected WLAN would not require any key transfers, as the WEP-key is pre-stored in every AP. However, neither roaming nor handover was specified in the original standard.

### 10.3 Overview on 802.11i

In June 2004, 802.11i has been adopted as the new security standard for the wireless LAN technology 802.11 [93]. It replaces the original security architecture WEP discussed in the last section. As opposed to WEP, the new standard 802.11i offers access control via mutual authentication between a MD and the network. It supports a large variety of authentication protocols that are implemented in combination with a key-agreement protocol.

\(^2\)RC4 is a stream cipher developed by Ron Rivest (see [66]).

\(^3\)Note that here, only a valid key stream for one initialization vector is revealed, but not the actual long-term secret key.

\(^4\)Note that all MDs use the same secret key as input to the RC4 key-stream generation. Thus, in order to recover the WEP-key, an attacker can use the intercepted traffic of several MDs and does not have to wait until he has intercepted enough traffic from a single MD.
In addition, it protects the confidentiality and integrity of the air interface between MD and AP. In the next sections, we describe the authentication and key-agreement protocols, the key establishment, the encryption and integrity-protection mechanisms, as well as the security-mechanism negotiation specified for an 802.11i-protected WLAN (see [56, 84] or the standard [93] for a detailed description of each of these mechanisms).

### 10.3.1 Authentication

802.11i supports two different types of authentication and key-agreement protocols: one that is based on a Pre-Shared Key (PSK) and one that is based on 802.1X [94].

The 802.1X-based authentication is a protocol between MD, the AP with which it associates, and AS, which controls the network access for one or more APs. MD first associates with an AP within its range using WEP's Open Authentication. The association only enables MD to exchange authentication data with AS. Any other traffic is blocked until the authentication is completed successfully. The standard does not specify any particular authentication protocol to be used between MD and AS. Instead, it specifies a WLAN-adapted implementation of EAP [36], on top of which different authentication methods can be used.

Figure 10.2 describes the protocol architecture between MD, AP, and AS. EAP method stands for the actual authentication mechanism used. Various EAP methods have already been defined, including EAP-TLS [14], EAP-TTLS [46], and EAP-SIM [88]. Different EAP methods can be based on different types of credentials. EAP-TLS is, for example, based on public-key certificates for MD, as well as network authentication. EAP-TTLS uses a public-key certificate for network authentication and a username/password combination for MD authentication. EAP-SIM is based on a shared long-term secret key. Only key-generating EAP methods can be used in connection with 802.11i. A key-generating EAP method generates a master key called Pairwise Master Key (PMK) in the 802.11i context. EAP itself is the end-to-end transport protocol for the EAP-method between MD and AS. EAPoL transports EAP over 802.x LANs and implements a port-based access control. Each association of MD with an AP creates a pair of IEEE 802.1X-controlled ports. Both sides implement a port blocking that blocks all traffic until the 802.1X authentication procedure completes successfully. RADIUS [150] can be used to transport EAP over IP to establish an authenticated channel between AP and AS, as well as to securely transport the generated key from AS to AP. The use of RADIUS is not required but suggested in the standard.

The PSK-based authentication is, as in WEP, implemented as a protocol between MD and the AP with which it wants to associate. However, as opposed to WEP, each AP has to store an individual key for each MD. In case of PSK-based authentication, the PSK is directly used as the master key (static key agreement $ka$).

---

5 To allow for a PSK-based authentication with central storage of the individual pre-shared-keys, the EAP method EAP-PSK has been proposed by Bersani et al. [31], in which the PSK for each MD is stored in an AS.
10.3 Overview on 802.11i

As described in the last section, if the 802.1X-based authentication is used, MD and AS generate a secret Pairwise Master Key PMK using an EAP method. Upon completion of the PMK generation, AS transfers the PMK to the AP MD is associated with. If the authentication is PSK-based, the PSK is used as the PMK directly. The PMK corresponds to the master key in our security model (see Chapter 1).

After any authentication and PMK agreement, AP and MD use the EAPoL-Handshake to generate a Pairwise Transient Key (PTK). The PTK is derived from the PMK, the MAC addresses of MD and AP, and from two nonces exchanged between MD and AP. PTK consists of three parts. The first part is used for key confirmation in the EAPoL-Handshake, the second part is used for encrypted transfer of the Group Transient Key (GTK) (used for broadcast traffic from AP to all associated MDs) and the last part is used as the Temporal Key (TK) for encryption and integrity protection of the subsequent MAC layer traffic. Figure 10.3 provides an overview on the 802.1X-based authentication, key agreement, key transfer and key establishment. Figure 10.4 describes the key hierarchy used in 802.11i.
10.3.3 Encryption and Integrity Protection

802.11i supports three encryption mechanisms, two of which come with an integrated integrity protection. The first one is the Temporal Key Integrity Protocol (TKIP) that is based on the stream cipher RC4 and an integrity-protection mechanism called Michael [181]. The second encryption and integrity-protection mechanism is AES-based and is called Counter Mode with CBC-MAC Protocol (CCMP). In order to offer backward-compatibility, 802.11i additionally supports WEP encryption as a third encryption method. Note that 802.11i allows for encryption and integrity protection to be disabled completely, in case it is allowed by both MD and the network.

10.3.4 Security-Mechanism Negotiation and Policies

In 802.11i, an AP advertises its network’s security policy, namely the allowed authentication and key-agreement protocols, as well as the allowed encryption and integrity-protection mechanisms, in the beacon messages it broadcasts. The selection of the security suite to use is left to the mobile device that requests association. MD sends its selection to the network as part of its initial association request. This corresponds to negotiation Method 2 described in Section 1.3.

In particular, standardwise the preferences of MD and HN are not respected on security-mechanism negotiation. However, as the standard does not specify how MD chooses from the security suites advertised in the beacons, MD may be set up in a way that its preferences may be respected during security-mechanism negotiation. If, in addition, HN would be able to advertise its security capabilities with preferences, MD may be able to take HN’s preferences into account.
preferences into account when reconciling the policies. In particular, MD can use Method 4
to reconcile HN’s and MD’s policies to its favor. Note that the standard does not allow
for an easy integration of the negotiation Method 5 presented in Section 1.3, as it does not
support negotiations taking several rounds.

10.3.5 Pre-Registration

The pre-registration process for a WLAN can take many different forms. In the non-
commercial case, a user typically registers with the system administrator of the WLAN
directly. In the commercial case, public WLAN providers sometimes require a user to sign
a contract similar to Mobile Phone Operators. Other public WLAN providers only require
their users to signup online.

Whatever form the registration process takes, a user and the network provider establish
some form of credentials during this process. These credentials can either be a PSK or any
other form of credential as required for the EAP method MD and the network are to use to
authenticate each other. To offer a variety of EAP methods to a user, a network provider
may even allocate and distribute different types of credentials to its users. By selecting
an authentication and key-agreement protocol on connection establishment, MD indirectly
determines the credential type it wants to use.

10.4 Conclusion

In this chapter, we have provided an overview on the security architecture 802.11i. We
briefly discussed authentication and key agreement, key transfer and establishment, encryp-
tion, integrity protection, and security-mechanism negotiation in 802.11i-protected WLANs.
With the introduced basic knowledge on 802.11i, we can now proceed with a description
of our new inter-provider roaming authentication protocol EAP-TLS-KS (Chapter 11) and
with the details on the new HEPB-approach for the WLAN case (Chapter 12). We refer to
[56, 84] or the standard document [93] for a detailed description of each of the mechanisms
introduced here.
Chapter 11

The New Protocol EAP-TLS-KS

As WLAN hotspots become widely available in airports, train stations, coffee shops and hotels, there is an increasing need for easy to use authentication protocols, which enable roaming between different Wireless Internet Service Providers (WISPs). Most WISPs currently use the web based Universal Access Method (UAM) for authentication, a method which is also recommended as the best current practice for inter-provider roaming by the Wi-Fi Alliance [18]. However, UAM is known to be vulnerable to many different attacks, such as impersonation of an AP, dictionary attacks, and service theft by means of address spoofing [178].

The new standard 802.11i [93] was put forward to address these problems. The use of MAC layer encryption between MDs and APs protects against service theft by means of address spoofing. Furthermore, 802.11i requires mutual authentication between MD and a network, thus protecting against fake APs. Every EAP-Method supported by 802.11i can be used to authenticate roaming MDs with the help of a hierarchy of ASs. Upon roaming to FN, MD presents its EAP-Identity to AS$_{FN}$. MD’s EAP-Identity includes an identifier of MD’s HN. AS$_{FN}$ forwards all authentication traffic between MD to AS$_{HN}$ until MD is successfully authenticated by AS$_{HN}$. After successful authentication AS$_{HN}$ transfers the generated PMK to AS$_{FN}$ which in turn forwards it to the currently serving AP. Every EAP-Method can thus be used as a roaming authentication protocol of Type2 (see Chapter 2). As discussed, before roaming authentication protocols of this type have the disadvantage of requiring a secure channel between AS$_{HN}$ and AS$_{FN}$ for key transfer.

In [29, 81] public-key-based authentication methods of Type1 were suggested. These suggestions share the general shortcoming of all public-key-based authentication methods for roaming users discussed in Chapter 2: MD must check the validity and revocation status of certificates during network authentication, i.e., before actually having network access. In [29] this problem is addressed by delegating certificate-chain discovery and validation to a trusted authority.

In this chapter, we solve the problems of state-of-the-art public-key-based roaming methods and the typical use of EAP-Methods to implement roaming between IEEE 802.11i-protected WLANs by adapting the secret-sharing approach introduced in Chapter 2.2 to
the WLAN case. In particular, we present a new protocol EAP-TLS-KS, which implements
the new concept based on EAP-TLS. This part of the thesis is joint work with J. Cordasco
and S. Wetzel and has been published in [121].

We show that due to the key-splitting approach EAP-TLS-KS allows for efficient certificate handling. In particular, MD does not need to validate any certificates upon roaming, as HN’s certificate is pre-installed on MD.

Furthermore, the use of secret sharing eliminates the need for a secure channel between FN and HN upon public-key-based roaming of MD. This is due to the fact that the splitting of the secret key allows FN to derive any necessary keying material itself.

EAP-TLS-KS is designed such that it differs from the original EAP-TLS protocol only on the server side. In particular, it is the public-key operations, which are performed by HN in EAP-TLS—such as decryptions and signatures—that are split between HN and FN in EAP-TLS-KS. We specify three protocol variant in order to support all types of EAP-TLS certificates. The first and second variants use conventional distributed RSA operations. For the third variant, we present a new distributed DSS signature scheme. This new scheme can generally be used in applications that exhibit an asymmetry in signing capabilities of the individual parties. That is, while one party (in our case HN) can generate valid signatures on its own, the second party (in our case FN) requires the other party’s cooperation in order to generate a valid signature. We also show that the new EAP-TLS-KS protocol has a performance advantage over EAP-TLS, as it reduces the number of round-trip message exchanges required between HN and FN. While the original EAP-TLS protocol needs four round-trip message exchanges between HN and FN, the new protocol requires only two.

Aside from addressing the security problems of current roaming solutions, in [121] we also show how key splitting can be used to efficiently support the fine-grained billing of a micropayment scheme. However, as accounting issues are out of the scope of this work, we do not detail this application of our new approach here.

Outline. In Section 11.1, we give an overview on EAP-TLS. In Section 11.2, we present the new protocol EAP-TLS-KS. We first adapt the key-splitting approach to the public-keys used in EAP-TLS and present two-party versions of the public-key operations used in EAP-TLS. In particular, we introduce a new method for two-party DSS signatures in Section 11.2.3. This is followed by a description and detailed discussion of the EAP-TLS-KS protocol including a security analysis. We close the chapter by summarizing related work in Section 11.3.

11.1 Overview of EAP-TLS

EAP-TLS is an EAP method defined in RFC 2716 [14] based on TLS [52]. It supports either mutual public-key certificate-based authentication or server authentication only. If EAP-TLS is used in 802.11i for server authentication only, another authentication method must be combined with EAP-TLS to implement client authentication. In this section, we describe the use of EAP-TLS with mutual certificate-based authentication. Figure 11.1 gives an overview of the EAP-TLS protocol and shows the encapsulation of TLS in EAP messages:
After agreeing upon the use of EAP, AS sends the client (in our case MD) an EAP-Request message requesting the client’s identity. The client answers with an EAP-Response message including its identity. AS then sends the TLS-Start message in an EAP-Request message and the TLS-Handshake begins:

The client sends the TLS-Client-Hello message in an EAP-Response message to the server. Client-Hello includes a random number (Client.RAND) that guarantees the freshness of the resulting keys to the client. The server answers with an EAP-Request message including the TLS messages Server-Hello, Server-Certificate, Client-Certificate-Request and Server-Done, and optionally the Server-Key-Exchange message. Server-Hello includes a random number (Server.RAND) that guarantees key freshness to the server. The certificate of the server is of one of the following three kinds:

1. A certificate including a public RSA key usable for encryption and signed by a CA with an RSA signature key (RSA).
2. A certificate including a public RSA key usable for RSA signature verification, signed by a CA with an RSA signature key (DHE-RSA).

3. A certificate including a public DSS key usable for DSS signature verification, signed by a CA with a DSS signature key (DHE-DSS).

EAP-TLS supports two methods for generating keying material. One is RSA encryption based (RSA case) and the other is based on a Diffie-Hellman key exchange (DHE case). In the RSA case, the server uses a certificate of type RSA and no `Server-Key-Exchange` is sent. In the DHE case, the server uses a certificate of type DHE-RSA or DHE-DSS and `Server-Certificate` is followed by `Server-Key-Exchange`. This message includes the server’s public DH value for this protocol instance. The hash value of the server’s public DH value concatenated with Client.RAND and Server.RAND is signed with the server’s RSA or DSS signature key and included in `Server-Key-Exchange`.

The client answers with `EAP-Response` including the TLS messages `Client-Certificate`, ..., `Finished` (see Figure 11.1). `Client-Certificate` is a DHE-RSA or a DHE-DSS certificate, depending on what type of certificate the server requests. `Client-Key-Exchange` is different for the RSA case and the DHE case:

1. In the RSA case, `Client-Key-Exchange` includes a random number (Sec.RAND) generated by the client and encrypted with the server’s public key.

2. In the DHE case, `Client-Key-Exchange` message includes the client’s public DH value.

In order to prove the client’s identity to the server, the client’s response includes `Client-Certificate-Verify`. This message contains a hash value of all messages sent and received so far starting from `Client-Hello` up to and including `Client-Key-Exchange` and is signed using the client’s signature key. The same `EAP-Response` message also includes `Change-Cipher-Specs` and `Finished`. With the former, the client indicates that it will now use the new ciphers and keys. The latter, which is protected with the new cipher-suite and keys, confirms that the client uses the same cipher-suite and keys as the server.

The server answers with the `Change-Cipher-Specs` and the `Finished` messages. By verifying the correct encryption of the `Finished` message, the client obtains a proof of the server’s identity since only the server can generate the correct session key. The client indicates successful receipt and verification by replying with an empty `EAP-Response` message. The EAP-TLS protocol ends with an `EAP-Success` message sent from the server to the client.

Figure 11.2 details the RSA case. The server certificate includes a public RSA encryption key. Therefore, `Server-KeyExchange` is not sent. `Client-Key-Exchange` consists of a random number Sec.RAND encrypted with the server’s public RSA key. Server and client generate the master secret by using Client.RAND, Server.RAND, and Sec.RAND as input to a Pseudo-Random Number Generator (PRNG). Figure 11.3 details the DHE case. `Server-Certificate` includes a public RSA or DSS signature-verification key.
11.2 EAP-TLS with Key Splitting

Applying the concept of key splitting to EAP-TLS, each HN is issued a roaming certificate that includes a public RSA encryption key, a public RSA signature-verification key, or a public DSS signature-verification key. HN splits and distributes the secret roaming key as described in Section 2.2:

Assuming HN has a pairwise roaming agreement with \( l \) foreign networks \( F_{N_1}, \ldots, F_{N_l} \), HN splits its secret roaming key \( \mathcal{R} \) into \( l \) different pairs of shares \((\mathcal{R}_{\text{HN}}, \mathcal{R}_{F_{N_i}})\) by means
of individual (2, 2) secret-sharing schemes with $\mathcal{R}_{HN_i} \neq \mathcal{R}_{HN_j}$ and $\mathcal{R}_{FN_i} \neq \mathcal{R}_{FN_j}$ for $i \neq j$. HN then distributes $\mathcal{R}_{FN_i}$ to FN$_i$ and keeps copies of $\mathcal{R}_{HN_i}$, as well as the secret roaming key $\mathcal{R}$. This not only allows HN to use the secret roaming key in case MD wants to access HN directly, but it also enables HN to issue suitable shares to new roaming partners. By construction, $\mathcal{R}$ can be recovered from a collection of shares, if and only if this collection includes a pair $(\mathcal{R}_{HN_i}, \mathcal{R}_{FN_i})$ for some $i \in \{1, \ldots, l\}$. In particular, $\mathcal{R}$ cannot be reconstructed from any pair $(\mathcal{R}_{HN_i}, \mathcal{R}_{FN_j})$ (with $i \neq j$) or any collection of shares of foreign networks only. Constructing key pairs with $(\mathcal{R}_{HN_i}, \mathcal{R}_{FN_i}) \neq (\mathcal{R}_{HN_j}, \mathcal{R}_{FN_j})$ for $i \neq j$ is necessary in order to allow for the unique identification of FN$_i$ upon successful authentication. Each MD stores the roaming certificate of its HN.

Upon roaming to FN, FN and MD initiate an EAP-TLS-KS authentication. MD acts like a regular client in the EAP-TLS protocol, as the EAP-TLS-KS protocol differs from EAP-TLS only on the server side. Depending on the type of roaming certificate, either RSA decryption, RSA signature generation or DSS signature generation is split between HN and FN using their respective shares of the secret roaming key. In the following sections, we detail the distributed schemes, as well as the key splitting for an RSA encryption key, an RSA signature key, and a DSS signature-verification key as the public roaming key.

### 11.2.1 Distributed RSA Decryption

The public roaming key is a pair $(n, e)$ of an RSA modulus $n = pq$ with $p, q$ prime and an RSA encryption key $e$ with $\gcd(e, \varphi(n)) = 1$, where $\varphi(n) = (p - 1)(q - 1)$. The secret roaming key $d$ is the inverse of $e$ modulo $\varphi(n)$: $ed = 1 \mod \varphi(n)$. Let $l$ be the number of FNs to receive a share of the secret roaming key from HN. Then, HN splits the roaming key $d$ into $d_{HN_i}, d_{FN_i}$ for $i = 1, \ldots, l$ implementing the access structure $\Gamma^1$ in the following way: HN randomly chooses $\omega_1, \ldots, \omega_l \in \mathbb{Z}_{\varphi(n)}$ such that $\omega_i \neq \omega_j$ for $i \neq j$.\footnote{$\Gamma = \{\{HN\}, \{\{HN_1, FN_1\}, \ldots, \{\{HN_l, FN_l\}\}\}$.} Then,

$$d_{HN_i} = d + 2\omega_i \mod \varphi(n), \quad i = 1, \ldots, l$$

$$d_{FN_i} = d + \omega_i \mod \varphi(n), \quad i = 1, \ldots, l.$$

Consequently, $d = -d_{HN_i} + 2d_{FN_i} \mod \varphi(n)$, for all $i = 1, \ldots, l$. HN distributes $d_{FN_i}$ to FN$_i$ and keeps copies of all $\omega_i$ as well as $d$.

Since $\omega_i \neq \omega_j$, each FN gets a different share. Since additionally $0 \leq \omega_i \leq \frac{\varphi(n)}{2}$, it holds that $2\omega_i \neq 2\omega_j$ for $i \neq j$. Thus, HN keeps a different share for each FN. The pair of shares $(d_{HN_i}, d_{FN_i})$ for a foreign network FN$_i$ thus uniquely carries the identity of FN$_i$.

HN and any single FN$_i$ can now decrypt a message $m$ encrypted to $c = m^e$ with the public encryption key $e$ in a distributed way: HN first computes $e^{-d_{HN_i}}$, and sends the result to FN$_i$. FN$_i$ then computes $e^{-d_{HN_i}}e^{2d_{FN_i}} = e^{-d_{HN_i} + 2d_{FN_i}} = c^d = m$.

**Security Analysis.** No FN can decrypt any message encrypted with $e$ on its own, as its knowledge of the share does not reveal any information about the decryption key $d$.\footnote{Here, $\mathbb{Z}_x$ denotes the set of residues modulo $x$.}
Furthermore, by construction, no pair or any larger coalition of FNs learn anything about the secret key from combining their key shares. An attacker intercepting the message $c^{-d_{HN}}$, sent from HN to FN, does not obtain any information on the plaintext $m$. Otherwise, the attacker would be able to break an RSA encryption with the public key $-ed_{HN}$, which contradicts the RSA assumption. Likewise, an attacker does not gain any information on $d_{HN}$ from his knowledge of $c^{-d_{HN}}$. We refer to [38] for a formal security analysis of additional properties of this distributed RSA decryption scheme.

### 11.2.2 Distributed RSA Signatures

The public roaming key is a pair $(e, n)$ of an RSA modulus $n = pq$ with $p, q$ prime and RSA signature-verification key $e$ with $\gcd(e, \varphi(n)) = 1$. The secret roaming key $d$ is the inverse of $e$ modulo $\varphi(n)$. The key splitting works exactly as in the RSA encryption-based case previously described. HN, together with any one of the foreign networks FN, can sign the hash value $h(m)$ on a message $m$ in the following distributed way: HN partially signs $h(m)$ by computing $D_{d_{HN}}(h(m)) = h(m)^{-d_{HN}}$, and sends the result to FN. FN computes $s = h(m)^{-d_{HN}}h(m)^{2d_{FN}} = h(m)^{-d_{HN}} + 2d_{FN} = h(m)^d$. Upon receipt of $s$ and $m$, MD can check the signature by verifying that $h(m) = s^e \mod n$.

**Security Analysis.** By construction, knowing only its share does not allow FN to sign a hash value by itself. Furthermore, no pair or larger coalition of FNs learn anything about the secret key by pooling their key shares.

An attacker intercepting the partially signed hash value $h(m)^{-d_{HN}}$, sent from HN to FN, cannot complete the signature without knowledge of $d_{FN}$. If an attacker could generate a valid signature $h(m)^d$ on $h(m)$ from $h(m)^{-d_{HN}}$, he could compute $h(m)^{2d_{FN}} = h(m)^{d-d_{HN}}$, and thus generate valid RSA signatures for a secret key $d_{FN}$. Thus, the distributed signature scheme is as secure as the original RSA signature. For a formal analysis of additional security properties, we refer to [114].

### 11.2.3 New Distributed DSS Signatures

The public roaming key is a DSS signature-verification key $(p, q, \alpha, y)$, where $p$ and $q$ are primes, $q|\langle p - 1 \rangle$, $\alpha \in \mathbb{Z}_p$, $\text{ord}(\alpha) = q$, $y = \alpha^a \mod p$.

The secret roaming key is $a$, which is randomly chosen from $\{1, \ldots, q - 1\} =: \mathbb{Z}_q^*$. The signature generation for the hash value $h(m)$ of a message $m$ for non-distributed signatures works as follows: The signer chooses a fresh $k^{-1} \in \{1, \ldots, q - 1\}$ for each signature and computes

$$r = \alpha^{k^{-1}} \mod p \mod q$$
$$s = k(h(m) + ar) \mod q.$$
The signature on $h(m)$ then consists of the pair $(r, s)$. For a more detailed description of the DSS signature generation and verification, see [120].

For a distributed DSS signature, it is necessary to split both the secret key $a$ and the ephemeral key $k$ between $HN$ and $FN_i$. In our new signature scheme, $HN$ splits the secret key $a$ for each $i = 1, \ldots, l$ multiplicatively into two parts $a_{FN_i}$ and $a_{HN_i}$. It distributes $a_{FN_i}$ to $FN_i$ and keeps a copy of each pair of shares $(a_{FN_i}, a_{HN_i})$.

During signature generation, the ephemeral key $k$ is chosen in a distributed manner. That is, $FN_i$ contributes one part, $K_{FN_i}$, while $HN$ contributes two parts, $K_{HN}$ and $k_{HN}$. $K_{FN_i}$ is known to $FN_i$ only. $K_{HN}$ and $k_{HN}$ are known to $HN$ only. $K_{HN}$ and $K_{FN_i}$ combine to $k_{FN_i}$, which becomes known to both $HN$ and $FN_i$ during signature generation. The ephemeral key $k$ is the product of $k_{HN}$ and $k_{FN_i}$.

Without loss of generality, the multiplicative splits of $a$ are generated by first using an additive splitting in the exponent rather than directly splitting it multiplicatively. HN selects $x \in \{1, \ldots, q - 1\}$ randomly and chooses $\omega_1, \ldots, \omega_l$ randomly in $\mathbb{Z}_{q-1}$ with $\omega_i \neq \omega_j$ for $i \neq j$. Then,

$$x_{HN_i} = x + 2 \omega_i \mod q - 1$$
$$x_{FN_i} = x + \omega_i \mod q - 1.$$

Thus,

$$x = -x_{HN_i} + 2x_{FN_i} \mod q - 1$$

for all $i = 1, \ldots, l$. All generated shares are different: $x_{HN_i} \neq x_{HN_j}$ for $i \neq j$ and $x_{FN_i} \neq x_{FN_j}$ for $i \neq j$. HN defines $a = \alpha^x \mod p \mod q$ and

$$a_{HN_i} = \alpha^{-x_{HN_i}} \mod p \mod q, \quad i = 1, \ldots, l$$
$$a_{FN_i} = \alpha^{2x_{FN_i}} \mod p \mod q, \quad i = 1, \ldots, l$$

such that $a_{HN_i} \cdot a_{FN_i} = \alpha^{-x_{HN_i}} \cdot \alpha^{2x_{FN_i}} = \alpha^x = a \mod p \mod q$. HN together with any FN can now generate a distributed DSS signature as follows: $FN_i$ first chooses $K_{FN_i}$ randomly from $\{0, \ldots, q - 1\}$ and sends $\alpha^{K_{FN_i}}$ to $HN$. HN then chooses $K_{HN}, k_{HN}, R_{HN}$ and $R_{FN_i}$ randomly in $\mathbb{Z}_q^*$ and computes $k_{FN_i}^{-1} = (\alpha^{K_{FN_i}})^{K_{HN}} \mod p \mod q$. HN ensures that $k = k_{FN_i}k_{HN_i} \mod q$ has not yet been used with the same secret key $a$ before. Then, $HN$ computes $r = \alpha^{k_{HN}^{-1}k_{FN_i}^{-1}}$, and

$$s_{HN} = (k_{HN} - R_{HN})k_{FN_i} \cdot h(m) + (k_{HN} \cdot a_{HN_i} - R_{FN_i}^{K_{HN}})k_{FN_i} \cdot a_{FN_i} \cdot r$$

$$= k_{HN} \cdot a_{FN_i} \cdot h(m) + k_{HN} \cdot k_{FN_i} \cdot a_{FN_i} \cdot a_{HN_i} \cdot r$$

$$= - R_{HN} \cdot k_{FN_i} \cdot h(m) - R_{FN_i}^{K_{HN}} \cdot k_{FN_i} \cdot a_{FN_i} \cdot r$$
HN sends $\alpha^{K_{HN}}, r, s_{HN}, R_{HN}$ and $R_{HN}^* \text{ to } FN_i$. FN_i determines $k_{FN_i}^{-1} = (\alpha^{K_{HN}})^{K_{FN_i}}$ and

$$s_{FN_i} = k_{FN_i} \cdot R_{HN} \cdot h(m) + k_{FN_i} \cdot R_{HN}^* \cdot a_{FN_i} \cdot r.$$ 

Now FN_i can compute the signature part $s$ on $h(m)$ as $s = s_{FN_i} + s_{HN}$. The pair $(r, s)$ is now a valid DSS signature on the hash value $h(m)$ with $a = a_{HN_i} \cdot a_{FN_i}$ and $k = k_{FN_i} \cdot k_{HN}$.

Thus, it can be verified by MD in the same way as a non-distributed DSS signature with ephemeral key $k$ and secret key $a$.

**Security Analysis.** By construction, FN cannot generate a valid signature on its own as its share does not provide any information on the key $a$.

An attacker cannot generate $s_{FN_i}$ without knowledge of $k_{FN_i}$ and $a_{FN_i}$. This is due to the fact that $s_{FN_i}$ is indeed a DSS signature on $R_{HN} \cdot h(m)$ with ephemeral key $k_{FN_i}$ and long-term key $a_{FN_i} \cdot R_{HN}^*$. If DSS is secure against existential forgery, then it is not possible to generate $s_{FN_i}$ without knowledge of $k_{FN_i}$ and $a_{FN_i} \cdot R_{HN}^*$. This is equivalent to the knowledge of $k_{FN_i}$ and $a_{FN_i}$, as $R_{HN}^*$ is public.

Two or more collaborating FNs cannot generate a valid signature, as they cannot reconstruct the secret key $a$ from their shares.

From intercepting $\alpha^{K_{HN}}, r, s_{HN}, R_{HN}, R_{HN}^*$, and $(s, r)$, an attacker cannot derive any information on $a$, $a_{HN_i}$, or $a_{FN_i}$.

HN chooses its contribution to $k$ without revealing it to FN. HN makes sure that no value of $k$ is used twice to generate a signature. FN or any attacker that interferes with the DH exchange used to exchange $k_{FN_i}$ thus cannot force the same $k$ to be used twice.\(^7\)

Unlike in the two RSA cases discussed previously, in the DSS case, HN uses its knowledge of the shares of the FNs during the signature-generation process. As discussed earlier, it is HN that should control authentication, as it will be the entity responsible for accounting. Nevertheless, it is important to note that the discussed distributed version of DSS is restricted to areas of application where one of the signers can sign on its own, while the other will need the cooperation of the first party to generate signatures.

Section 11.3 will provide a detailed discussion on how our distributed DSS signature scheme differs from previous work in the area.

### 11.2.4 The Protocol

#### 11.2.4.1 RSA Encryption-Based Key Generation

Figure 11.4 describes the EAP-TLS-KS protocol in the case where an RSA encryption key is used as the public roaming key and the key generation is based on RSA encryption. The protocol starts with the regular EAP-Request-Identity and EAP-Response-Identity messages, and the EAP-TLS-Start message between FN’s authentication server and MD. After receiving MD’s identity, FN sends an EAP-TLS-KS-Start message to HN. This message includes the client’s EAP-ID, as well as the identity of FN. All of the following EAP-TLS messages are encrypted using the client’s public key, which is stored in the EAP-TLS-KS-Start message.

\(^7\)It is well-known that in DSS, using the same ephemeral key $k$ twice with the same secret key $a$ reveals $a$ [120].
Figure 11.4: EAP-TLS-KS with RSA
messages, starting with Client-Hello and ending with Client-Certificate-Verify, are forwarded between MD and HN by FN.

In order to acknowledge FN’s identity to MD, HN includes FN’s identity in any of the TLS messages it sends to MD. In order to avoid any changes to the EAP-TLS implementation for MD, we integrate FN’s identity (FN-ID) as an attribute into the roaming certificate. MD can thus store the roaming certificate and refer to FN’s identity at any time. Upon receipt of the Server-Certificate message, which includes the roaming certificate, MD checks that the roaming key in the certificate matches the pre-installed key.

If they are equal, then MD chooses a random number Sec.RAND and encrypts it under the public roaming key \( e \). MD computes the hash value of all messages from the Client-Hello up to the Client-Key-Exchange messages, signs the computed hash value, and includes it in the Client-Certificate-Verify message. MD then sends the Client-Certificate, Client-Key-Exchange, Client-Certificate-Verify, Change-Cipher-Specs, and the Finished messages to FN.

FN forwards the messages Client-Certificate, Client-Key-Exchange, and Client-Certificate-Verify to HN. HN verifies MD’s certificate and its revocation status, as well as the signature on the Client-Certificate-Verify message. The correctness of the signature proves to HN that MD is in possession of the secret key corresponding to the public key in MD’s certificate. It furthermore proves that none of the messages exchanged between MD and HN so far have been altered in any way. In particular, this proves that MD received the same FN-ID as an attribute in the roaming certificate that HN sent. Thus, HN knows that both HN and MD associate the same identity with FN.

If the signature verification is successful, HN sends the partially decrypted random number \((\text{Sec.RAND})^e)^{i_{HN}}\), to FN. The receipt of this message assures FN that HN has successfully authenticated MD and thus authenticates MD indirectly to FN. FN fully decrypts Sec.RAND by computing \((\text{Sec.RAND})^e)^{i_{HN}} \cdot \text{Sec.RAND} = \text{Sec.RAND}^2\). FN can now compute the secret master key PMK = PRNG(Client.RAND, Server.RAND, Sec.RAND).

The Change-Cipher-Spec message indicates that the sending party will now switch to encryption mode. The Finished messages exchanged between FN and MD are encrypted with the secret master key PMK. By verifying that the Finished message it received is correctly encrypted, MD is assured that FN was able to generate the correct key. MD is furthermore assured that HN participated in the authentication and that the identity FN claimed to HN is correct and corresponds to the one included as an attribute in the roaming certificate.

### 11.2.4.2 DHE-RSA Case

Figure 11.5 describes the changes in the EAP-TLS-KS protocol in case an RSA signature key is used as the public roaming key and this key is used to sign the Server-Key-Exchange message, which includes the ephemeral DH key part (Server-Pub-DH) generated by FN and sent to HN as part of the EAP-TLS-KS-Start message. Upon receipt of the Client-Hello
message, HN computes the hash value
\[ h(m) = h(\text{Client.RAND}||\text{Server.RAND}||\text{Server-Pub-DH}) \]
and partially signs it to \( h(m)^{-d_{HN}} \). HN then constructs the \textbf{EAP-Request} message including \textbf{Server-Hello}, \textbf{Server-Certificate} (with the RSA public encryption key as roaming key and FN’s identity as attribute), \textbf{Server-Key-Exchange}, \textbf{Client-Certificate-Request}, and \textbf{Server-Done}. HN includes the partially signed hash in \textbf{Server-Key-Exchange}.

Upon receipt of this message, FN completes the RSA signature on \( h(m) \) by computing
\[ s = h(m)^{-d_{HN}} \cdot h(m)^{2d_{FN}}, \]
It then replaces the partially signed message with \( s \) in the \textbf{Server-Key-Exchange} message and forwards the information to MD. MD checks that the public-key in the roaming certificate is the one it has pre-installed. It generates its own public and secret ephemeral DH key values, computes the DH key from its own private DH value and the server’s public DH value, and generates the master key. MD then sends the respective EAP-TLS message to FN, including its public DH value in the \textbf{Client-Key-Exchange} message.

HN verifies MD’s certificate and MD’s signature on \textbf{Client-Certificate-Verify}. Note that HN has to generate the complete signature on the hash value included in the \textbf{Server-Key-Exchange} message and use it to replace the partially signed hash sent previously before it can compute the hash value of all messages sent and received so far. Otherwise, the signature will not be correct, as MD received the complete signature (and not just the partially signed message) included in the \textbf{Server-Key-Exchange} message from FN.

If HN can verify MD’s signature, then HN sends the \textbf{EAP-TLS-KS-Success} message to FN. The receipt of this message assures FN of the correctness of MD’s identity. FN now generates the DH key and the master key and eventually completes the EAP-TLS-KS protocol as in the original EAP-TLS protocol. As in the RSA encryption-based case, MD is assured of the correctness of FN’s identity by the correctness of the encryption of the \textbf{Finished} message received from FN.

### 11.2.4.3 DHE-DSS Case

Figure 11.6 describes the changes in the EAP-TLS-KS protocol when a DSS signature-verification key is used as the public roaming key. HN and FN jointly sign the server’s public DH key part using the distributed DSS signature described previously. The protocol is almost identical to the DHE-RSA case, except that FN has to send the ephemeral DSS key part \( \alpha^{K_{FN}} \) in addition to the Server-Public-DH value to HN in the \textbf{EAP-TLS-KS-Start} message. In both the DHE-based protocols, any FN can request HN to partly sign DH ephemeral keys regardless of any MD requesting to access FN. However, this can be noticed by HN if no \textbf{Client-Certificate-Verify} message follows the \textbf{Server-Key-Exchange} message. Moreover, as the hash value of the ephemeral DH key that is signed with the DSS or RSA key includes the Server.RAND and the Client.RAND, FN cannot use the recovered partially signed DH ephemeral key to fool MD.
EAP-TLS with Key Splitting

Figure 11.5: EAP-TLS-KS with DHE-RSA
Figure 11.6: EAP-TLS-KS with DHE-DSS
11.2 EAP-TLS with Key Splitting

We analyze the new protocol in comparison to two other usage scenarios of EAP-TLS for inter-provider roaming. The first one is to use the authentication server in HN and having FN act like an access point in the regular EAP-TLS protocol. This scenario is equivalent to the one depicted in Figure 11.1 when replacing the client with MD, the server with the authentication server in HN, and placing the authentication server of FN in the middle to forward all traffic between MD and HN’s AS. Here, we refer to this protocol as EAP-TLS (HN). Note that this setting requires a secure channel between FN and HN in order to transfer the master key from HN to FN.

In the second scenario, the authentication is fully delegated to FN, i.e., full control is delegated from HN to FN. This is equivalent to Figure 11.1 when replacing the server with FN’s authentication server. Here, we refer to this scenario as EAP-TLS (FN).

Table 11.1 compares the three protocols EAP-TLS (HN), EAP-TLS (FN), and EAP-TLS-KS in terms of the number of EAP messages sent, signatures generated and verified, and the number of messages encrypted or decrypted by MD, HN, or FN. The three key generation types are listed separately.

The number of EAP messages MD has to send, as well as the number of public-key operations MD has to perform, is the same for all three protocols. Compared to the EAP-TLS (FN) protocol, the new protocol has the advantage that MD can authenticate FN without having to check the validity of any chain of certificates and the revocation status of these certificates (* in Table 11.1). Instead, MD can use the pre-installed roaming certificate and simply check whether it matches the received server certificate. If the certificate is revoked due to compromise, HN must immediately notify MD. In case the certificate expired, MD trusts HN not to engage in the authentication. This addresses an important problem
with the other schemes where MD is required to check the validity of FN’s certificate itself or delegate this task to a trusted third party and wait for the respective response.

In comparison to EAP-TLS (FN), the new protocol shifts part of the load from FN to HN. In the new protocol, FN has to forward three EAP messages between MD and HN and send one additional EAP-TLS-KS message to HN. The load of verifying MD’s signature on the Certificate-Verify message and the load of validating MD’s certificate is shifted to HN.

Compared to EAP-TLS (HN), the new protocol significantly reduces the number of messages FN has to forward between MD and HN. The new protocol requires the forwarding of only three EAP messages, as opposed to the nine messages in EAP-TLS (HN). The new protocol, however, requires two additional EAP-TLS-KS messages to be exchanged between HN and FN, as well as an additional public-key operation by FN. Unlike in EAP-TLS (HN), the new protocol does not require any secure channel in order to transfer the master key from HN to FN.

11.2.5 Properties of EAP-TLS-KS

**Complete Control by HN.** HN fully controls every access requested by any of its subscribed MDs to any foreign network. HN furthermore fully controls the revocation status of both the roaming agreement with each FN as well as that of any MD. It can thus ensure that no successful authentication can take place after revocation. This eliminates the trust HN has to put into FN in other protocols, namely the trust that FN correctly checks the revocation status of MD’s certificate before granting access.

**Proof of FN’s ID to MD and Authentication of MD to FN.** As HN uses a different key share for every FN, MD gains an indirect proof of FN’s identity upon successful termination of the EAP-TLS-KS protocol. This proof of FN’s identity enables MD to configure its own roaming policy locally, e.g., by excluding certain networks or keeping preference lists. This furthermore allows for a simple integration of an accounting initialization into the authentication procedure. Details on this topic are discussed in [121].

Since authentication of roaming MDs requires HN to validate the certificate status, FN is assured that upon successful completion of the protocol, HN approves MD to use FN’s services and HN is willing to reimburse FN for providing service to MD.

**Elimination of Secure Channel between HN and FN.** As opposed to many other schemes (e.g., [118, 156]), the new authentication protocol does not require the existence of a secure channel between HN and FN. This is due to the key splitting, which makes it unnecessary to transfer the master key from HN to FN.

**Simple Integration of New Roaming Agreements.** A new roaming agreement between HN and a new foreign network FN_{l+1} requires the generation of a new pair of shares of the secret roaming key. In the RSA cases, HN simply generates two new shares of the secret roaming key d, namely \( d_{HN_{l+1}} \) and \( d_{FN_{l+1}} \), distributes \( d_{FN_{l+1}} \) to \( FN_{l+1} \). In the DSS case, it generates a new pair of shares of the secret roaming key a, namely \( (a_{HN_{l+1}}, a_{FN_{l+1}}) \). It keeps both shares to itself and distributes \( a_{FN_{l+1}} \) to \( FN_{l+1} \). In both cases, neither does
the provider have to change the roaming key pair nor does it have to update or change any of the already distributed shares. Our scheme thus accommodates expansion to an arbitrary number of roaming agreements. The security of the scheme does not depend on the number of FNs with which the home provider has roaming agreements. No adjustments are necessary for MD in order to allow for successful authentication to a new FN upon roaming.

**Efficient Revocation of Agreements and Subscriptions.** In order to revoke the roaming agreement with FN, HN simply marks the respective shares for that FN as revoked. Incoming authentication requests for the revoked FN are then no longer co-signed or partially decrypted by HN. There is no need for HN to change the public roaming key.

The revocation status of the certificates for MDs is maintained by HN. Consequently, no FN is required to check the status of MD. Instead, revocation of MD’s certificate is efficiently implemented by HN refusing to co-sign or partially decrypt authentication requests for revoked MDs.

**Simple Handling of Compromised Keys.** In case the key of a particular FN has been compromised, HN marks the corresponding shares as invalid. In particular, the compromise of the key share of an individual FN does not require the generating of a new roaming key pair. This is due to the fact that all FNs have individual shares.

In case the secret roaming key itself is compromised, HN has to immediately notify all its MDs of the revocation of the current roaming certificate and distribute a new one. However, it is not necessary for HN to provide the FNs with new shares. Consequently, the burden of expensive secret key-share distribution in case of a compromised key is eliminated. In the following paragraphs we discuss the details of handling compromised keys for the different roaming key types:

**RSA Cases.** The secret roaming key is a secret RSA key $d$ and the public roaming key is a public RSA key pair $(e, n)$. Let $d^*$ be the new secret roaming key to be split with the FNs and let $(e^*, n^*)$ be the corresponding public RSA key pair with $n^* \neq n$. Then, HN determines $\delta = d - d^* \mod \varphi(n^*)$ as well as $\delta_i = d + w_i - d_{FN_i} \mod \varphi(n^*)$ and replaces its own old shares with the new shares

$$d^*_{HN_i} = 2(w_i - \delta - \delta_i) + d^* \mod \varphi(n^*)$$

for $i = 1, \ldots, l$. Consequently, any pair $(d^*_{HN_i}, d_{FN_i})$ can now be used to reconstruct the new roaming key $d^*$ by:

$$-d^*_{HN_i} + 2d_{FN_i} = -2(w_i - \delta - \delta_i) - d^*$$

$$+ 2(w_i - \delta + d^* - \delta_i)$$

$$= d^* \mod \varphi(n^*)$$

In fact, the splitting of the new key $d^*$ is done in the same way as the splitting of $d$ by replacing $w_i$ with $w_i - \delta - \delta_i$. That is, the random contribution of $w_i$ is now provided by

---

8It is important to ensure that $n^*$ is different from $n$. Otherwise, since an attacker who knows $d$ can factor $n$ and thereby knows $\varphi(n)$ could also easily compute $d^*$ by inverting $e^*$ modulo $\varphi(n)$ in case $n^* = n$. 

\( \delta_i = d + w_i - d_{FN_i} \mod \varphi(n^*) \). If an attacker knows \( d \) and can thereby factor \( n \), he can also learn \( w_i \) by collaborating with \( FN_i \). However, the attacker cannot compute \( \delta \) or \( \delta_i \), as \( FN_i \) does not know \( \varphi(n^*) \). Even if two or more FNs collaborate, they cannot compute \( \delta \) as each collaborating \( FN_i \) increases the number of unknown variable by one (\( \delta_i \)).

**DSS Case.** The splitting of a new secret roaming key \( \alpha^* = a^* \) is obtained by determining a new additive splitting for \( x^* \). That is, \( HN \) chooses a new \( x^* \in \{1, \ldots, q-1\} \) and determines \( \delta = x - x^* \mod q - 1 \). It then computes

\[
x^*_{HN_i} = x_{HN_i} + \delta \mod q - 1
\]

and determines \( a^* = \alpha^{x^*} \) as the new secret key and \( a^*_{HN_i} = \alpha^{x^*_{HN_i}} \). Consequently, \( a^*_{HN_i} \cdot a^*_{FN_i} = \alpha^{-x^*_{HN_i} + 2x^*_{FN_i}} = \alpha^{x^*-\delta} = \alpha^{x^*} \). Thus, \( HN \) and \( FN_i \) now multiplicatively share the new secret key \( a^* \).

It is important to note that from \( a \), \( a_{HN_i} \), and \( a_{FN_i} \), the values \( x \), \( x_{HN_i} \), and \( x_{FN_i} \) cannot be recovered as long as the discrete logarithm assumption holds. Thus, if the key \( a \) is recovered by an attacker, \( x \), \( x_{HN_i} \), and \( x_{FN_i} \) are not affected. Consequently, the attacker learns nothing about \( a^* \).

An attacker who recovers \( a \) cannot use his knowledge on past signatures to compute new ones. As he knows nothing about the key \( a^* \), he is in exactly the same situation as any signer with a secret key that tries to forge signatures of another signer using the same public parameters. If the attacker could use his knowledge on past signatures to compute valid signatures without knowledge of \( a^* \), each signer could use his own past signatures to sign on behalf of someone else in DSS. The new splitting is therefore as secure against signature forgery as the original DSS signature scheme.

**Simple Update of Keys.** Updating of keys can be done in the same way as in the case of compromise.

### 11.3 Related Work

#### 11.3.1 Inter-Provider Roaming in Public WLANs

**Overviews on Inter-Provider Roaming.** In [24], Balachandran et al. discuss open questions and challenges related to WLAN hotspot providers with an emphasis on roaming issues and security. In [178], Wang et al. discuss and analyze the security mechanisms UAM, 802.1X, PANA, and USIM-based authentication for wireless hotspot providers and inter-provider roaming.

**Web-Based Authentication Methods.** The most widely-used authentication protocol by WISPs is the web-based universal access method UAM. This method has been shown to have several vulnerabilities [178]. A renegade AP connected to a web server with a valid SSL certificate can be set up in the hotspot and trick users into divulging their authentication credentials. Furthermore, UAM is vulnerable to dictionary attacks. A malicious MD can spoof the address pair (MAC/IP address) of an already authenticated MD to conduct
11.3 Related Work

service theft. Another web-based authentication method is CHOICE [23], which is secure against address spoofing. However, it uses the MS-Passport technology [126], which makes it platform dependent. A proprietary security sublayer between the link-layer and the IP layer further restricts the application area of CHOICE. In contrast, the protocol proposed in this chapter is based on publicly available technologies only. Spinach [19] offers a web-based interface to a Kerberos authentication service and aims to not only protect public wireless access points, but also to secure public network ports. It is designed to be flexible with respect to the use of other authentication methods if desired. Unfortunately, it is vulnerable to address spoofing.

Other Proposals. WLANs that support the new 802.11i security architecture can be set up to support any EAP method in combination with a RADIUS server proxy hierarchy [189] on roaming. FN’s authentication server acts as a proxy for all EAP-method-specific authentication traffic between MD and its home provider. The home provider’s authentication server authenticates MD in the same way as if MD requested service to it. Depending on the EAP method used the authentication can then be based on public-key certificates or be non-public-key based.

In [119], McCann et al. suggest enhancements to the two most widespread current roaming authentication methods used in WLAN: UAM and 802.11i-based authentication. UAM is enhanced to support other types of credential than username/password combinations in order to reduce the required user interaction to opening a web-browser. 802.11i-based authentication is in turn enhanced by a web-interface similar to UAM to facilitate migration from UAM to 802.11i for UAM-acustomed users.

Salgarelli et al. [156] suggest an authentication protocol EAP-W-SKE that minimizes the number of round-trip message exchanges between FN and HN to one round-trip. However, this requires a secure channel between FN and HN for key transfer. In [118], Matsunaga et al. propose a single sign-on authentication architecture that is based on 802.1X and EAP-TLS for PKI-based network authentication. It can be combined with any web-based authentication method for MD authentication. Due to the use of 802.1X, their architecture is secure against address spoofing. Yet, the web-based user authentication mechanisms require a secure channel between the local web server and the user’s identity server of choice. Our protocol does not require the existence of any secure channel between any of the components of the visited FN and HN. Moreover, in [118] it is assumed that MDs can check the validity of any public-key certificate presented by FN as part of the EAP-TLS protocol. The problems that arise from certificate-chain discovery and validation by MD are not addressed in [118].

Some protocols have been suggested for inter-provider roaming that use public-key-based methods to authenticate both the network and MD. Gu et al. make an explicit suggestion for WLAN [81], whereas Bayarou et al. suggest a general framework for wireless networks [29].

The authentication protocol in [81] shares the aforementioned deficiencies concerning costly discovery, verification, and validation of certificate chains by an offline MD. In [29], these shortcomings are addressed by delegating certificate-chain discovery and validation to a trusted server. This solves the offline problem but causes additional round-trips to the
trusted server. In [110], Long et al. describe an authentication and key-agreement protocol for roaming WLAN subscribers. The protocol is based on individual public-key certificates and the SSL protocol. Each home provider issues certificates for himself as well as his pre-registered users. Moreover, the home provider signs the individual certificate of each foreign provider he has a roaming agreement with. MD and FN authenticate each other using these certificates in a modified SSL handshake. The home provider is not engaged in the authentication. The authors additionally suggest that FN may present part of the SSL-handshake transcript as evidence of service provisioning to MD’s home provider for accounting purposes. A transcript, however, does not prove any usage time to the home provider. In particular, FN could require MD to authenticate several times in a row in order to obtain different transcripts and then be paid. Moreover, the protocol does not address how changes in roaming profiles and agreements are to be handled. In particular, no suggestion is made as to how a roaming agreement can be revoked. Revocation-status checking of user certificates is circumvented by restricting the lifetime of MD certificates to a month and assume fixed monthly payments to the home provider, thus restricting the risk of fraudulent use of a revoked certificate to the period of one month. In [149], Ribeiro et al. describe an IPsec-based roaming authentication approach that uses a hierarchy of certificate authorities to secure a non-commercial WLAN. This approach comes with two typical problems of IP-layer-only authentication protocols: an attacker can spoof authenticated (IP address, MAC address) pairs and can analyze traffic intercepted on the air interface based on the unencrypted IP headers.

11.3.2 Distributed Signatures

Secret-Sharing Scheme. For the construction of a secret-sharing scheme that represents our non-threshold composite access structure, we make use of an approach presented and proved in [117]. The authors show that a composite access structure \( \Gamma_0 \) allows for a vector-space construction if the initial access structure \( \Gamma_0 \) itself allows a vector-space construction. The proof is constructive and we use it in a straightforward manner to construct our linear secret-sharing scheme.

Geer and Yung suggest alternative applications of threshold cryptography in [71]. Although this work does not address inter-provider roaming, this paper inspired our work.

Distributed RSA. Distributed RSA decryption and signatures were first suggested and analyzed in [40] and [68]. We use these methods in EAP-TLS-KS in a straightforward manner. In [38], Boneh et al. use a semi-trusted mediator in conjunction with two-party RSA signature schemes and cryptosystems to facilitate the revocation of user certificates. We refer to their security analysis for the distributed RSA decryption. In [114], MacKenzie et al. use distributed RSA signatures to secure PIN-protected or password-protected private keys against off-line dictionary attacks in order to achieve capture resilience. We refer to their formal security analysis for the distributed RSA signature scheme.

\[^9\text{This corresponds to the third column of Table 2.1.}\]
11.4 Conclusion

In this chapter, we have presented a new protocol EAP-TLS-KS for authentication and key agreement on WLAN inter-provider roaming. Our solution addresses the most significant problems of current public-key-based approaches. It solves the problems related to certificate validation on MD by using only one pre-installed roaming certificate per home provider. Our protocol allows for flexible handling of changes in roaming agreements and profiles by guaranteeing full control over each roaming instance to HN. At the same time, our protocol reduces the number of round-trip message exchanges required between FN and HN upon roaming to two round-trips and is thus more efficient than the standard usage of EAP-based inter-provider roaming. Furthermore, our protocol does not require a secure channel between FN and HN, as the secret master key used to protect the air-interface between MD and FN after successful authentication is derived by FN itself.

Our protocol does not explicitly support anonymous roaming nor does it support quality-of-service dependent payment. Future work will explore how current research on these topics, such as the ideas in [155], can be integrated into our protocol. Generalizing the key-

Distributed DSS. Distributed DSS signatures are particularly hard to construct since not only a long-term secret key, but also the ephemeral key, has to be split between signers. A fully symmetrical two-party DSS signature-generation scheme was presented by MacKenzie et al. in [115]. This scheme requires a semantically secure public-key encryption scheme to be implemented between the two signers that exhibits a specific homomorphic property as, e.g., in the cryptosystems of Pallier [139] or Okamoto et al. [134]. The purpose of the encryption scheme is to allow one party to reveal his share encrypted with its public key to the other party. The other party can use the encrypted share to generate an encrypted signature and then send this back to the first party, who finally decrypts the full signature. The encryption scheme thus enables the symmetry of the two-party signature. We assume an asymmetric setting in which HN keeps the complete key as well as FN’s shares for several reasons. First, it allows HN to engage in new roaming agreements. Second, it enables simple key update and key-compromise handling. Finally, it allows HN to use the full secret key if MD “roams” to HN. The use of MacKenzie et al.’s distributed DSS version in our setting would thus generate unnecessary overhead. In contrast, the distributed DSS signature scheme introduced in this chapter is tailored to the specific setting of roaming in which one of the signing parties has complete power over the other. In the first work on distributed DSS signatures [107], Langford presented a \((2, l)\) threshold DSS signature for \(l \geq 3\). This construction was generalized by Gennaro et al. to a \((t, n)\) threshold signature with \(n \geq 2t + 1\) in [73]. These schemes are not directly applicable in our scenario, as we require a \((2, 2)\) signature scheme. However, our distributed version is similar to the \((2, 3)\) threshold DSS signature presented in [107]. Langford blinds the shares of each party with random numbers and uses a third party to unblind and finally compose partly signed messages into fully signed ones. We use a similar blinding with random numbers to conceal HN’s share from FN.
splitting approach to support roaming mediators, in addition to pairwise roaming agreements, is another interesting direction for future work.
Chapter 12

History-Enriched Policy-Based SCT for WLAN

Handover procedures within a WLAN network are standardized by the IEEE in [91, 93] and make use of the Inter Access Point Protocol IAPP [92]. The standardized procedures, however, do not support handover across different IP domains and consequently do not support inter-provider handover, as IAPP is restricted to APs connected to the WLAN.

The Context Transfer Protocol (CXTP) [112] and the Candidate Access Router Discovery (CARD) [108] are experimental protocols standardized by the IETF. In combination with Mobile-IP for mobility management, CXTP and CARD aim to support handover across IP domains.

In this chapter, we discuss how the history-enriched, policy-based security-context transfer can be used to enhance inter-provider handover across different WLANs. In this context, we show how the HEPB approach can be implemented using CARD and CXTP.

Implementing SRC-controlled HEPB security-context transfers using CXTP such that the security requirements R*-1 to R*-4, R*-7, and R*-8 are met, requires only minor changes to CXTP. It is, however, difficult, to implement the negotiation of the cipher suite in CXTP alone, such that the requirements R*-6 and R*-5 are met. In particular, in WLAN a mobile device chooses the cipher suite to be used from the capabilities advertised by the APs and sends its choice to the AP already in the association request. Therefore, we suggest integrating the negotiation of the security mechanisms to use after handover with CARD, such that the negotiation takes place before MD associates with an AP of DEST_k. The enforcement of MD’s, SRC_k’s, and DEST_k’s policies, however, is completed with the help of CXTP.

Another difficulty of integrating the HEPB approach in CXTP and CARD is that CXTP in its current version only supports context transfers from the access router in SRC_k to the access router in DEST_k. Consequently, CXTP cannot be used directly to implement HN-controlled or AN-controlled handover. In Section 12.4.2, we discuss the required changes to CXTP in order to accommodate HN-controlled and AN-controlled handover.

Outline. In Section 12.1, we map the context history, security context, policy, and handover
procedures of our model to the WLAN case. In Section 12.2, we give an overview of CARD, followed by an overview on CXTP in Section 12.3. In Section 12.4, we discuss the integration of the HEPB approach with CARD and CXTP. Related work on inter-provider handover between WLANs is discussed in Section 12.5, which is also the conclusion of this chapter.

12.1 HEPB Handover in the WLAN Context

In this section, we map the components of the HEPB procedure with key derivation introduced in Section 5.1 to IEEE 802.11i-protected WLANs.

Initial Security Context. The initial security context $S_0$ consists of: (1) the authentication and key-agreement protocol, which is either PSK-based or an EAP method; (2) one of the three encryption and integrity-protection mechanisms WEP, TKIP, or CCMP. The key-establishment protocol in 802.11i-protected WLANs is pre-defined as the EAPoL-Handshake described in Section 10.3.2 and therefore is not included in the initial (or any other) security context.

Key Derivation. Upon handover, HCN derives the next pairwise master session key $K_k$ from $K_0$ (HN-controlled and AN-controlled cases) or from $K_{k-1}$ (SRC-controlled case), as described in Section 5.1.2.5. As a key-derivation function, we suggest using the TLS Pseudo-Random Function (TLS-PRF) implemented on each MD that supports EAP-TLS. This function is based on SHA-1 and is currently assumed to be pre-image resistant, such that $R^*-2$ is met. As described in Section 5.1.2.5, our key-derivation methods meet $R^*-3$ only in part.

Context History and Security Context. The context history consists of the initial authentication and key-agreement protocol, as well as all subsequently used cipher suites $cs \in \{WEP, TKIP, CCMP\}$ and all previously used key-derivation functions. As described in Section 5.1.4, we omit the order or frequency of appearance in the context history. The history thus maximally includes all three cipher suites and the overall number of possible histories is

$$8 \cdot (|\text{defined EAP methods}| + 1) \cdot |\text{key-derivation functions}|.$$

The security context transferred during a $k$-th-order handover is

$$(K_k, history_{k-1}, Tr_{HCN}),$$

where $K_k$ is the pairwise master session key derived as described in the last paragraph.

Policies. MD, HCN, and DEST express their policies with respect to the cipher suites they allow to be used after handover by pre-defining sets of cipher suites dependent on the context histories. In order to support MDs and networks in pre-defining the sets of allowed cipher suites, a tool that translates policies of MD and the networks into certain sets of
cipher suites can be developed. Such a tool could, for example, allow an entity \( X \) to set cipher suite sets \( CS_X|_{ssh_{k-1}} \) to be empty if a certain initial authentication and key agreement is included in \( ssh_{k-1} \) or if a certain encryption and integrity-protection mechanism (e.g., WEP) appears in \( ssh_{k-1} \). It could set all other histories to a certain pre-defined subset of \( CS = \{\text{WEP, TKIP, CCMP}\} \).

**Procedure.** The procedures themselves are the ones already described in Section 5.1.3. However, an interesting question is how the history-enriched, policy-based context transfer can be implemented reusing other already standardized handover-supporting protocols. Consequently, in the rest of this chapter we discuss how the HEPB approach can be implemented using two recently developed experimental IETF protocols, CARD and CXTP.

### 12.2 The Candidate Access Router Discovery (CARD)

The Network Working Group of the IETF specifies the so-called Candidate Access Router Discovery (CARD) Protocol in RFC 4066 [108]. This protocol supports MD in resolving the IP address and certain other capabilities of so-called Candidate Access Routers (CAR) for handover available at MD’s current location.

In the context of WLAN, this works as follows. The mobile device periodically scans for candidate APs. The scanning results in the discovery of the MAC addresses of the APs available at MD’s current location. The CARD protocol allows a mobile device to request the access router with which it is currently connected to resolve the MAC addresses of the candidate APs to the IP addresses of the corresponding candidate access routers. Additionally, CARD allows MD to request its current access router for other capabilities of the candidate access routers, such as the available bandwidth that might influence its handover decision.

The CARD protocol can thus be used by MD to generate the ordered list of candidate destination access routers. In addition, CARD can be used to transfer sets of cipher suites allowed to be used after handover as part of the capabilities of the access routers. We detail this in Section 12.4.

### 12.3 The Context Transfer Protocol (CXTP)

The SEAMOBY Working Group [79] of the IETF specifies the so-called Context Transfer Protocol (CXTP) to support mobility in all-IP-based networks in RFC 4067 [112]. This protocol enables context transfers between access routers in different IP domains initiated by MD or the source access router. The protocol does not specify the actual context-transfer data, but standardizes a protocol to transfer different context types.

In the WLAN context, handover procedures are mobile-initiated. Consequently, we describe only the mobile-initiated part of CXTP here. CXTP supports only SRC-controlled context transfers. CXTP supports two types of mobile-initiated context transfers: **predictive** transfers and **reactive** transfers. Predictive transfers correspond to our SRC-controlled
mobile-initiated handover where SRC\(_k\) is notified by MD (see Figure 3.9), while reactive context transfer corresponds to the procedure where SRC\(_k\) is notified by DEST\(_k\) (see Figure 3.8). Predictive transfer takes place if MD detects a handover reason and is able to notify SRC\(_k\) before it loses connection to SRC\(_k\). In this case, SRC\(_k\) pushes the context to DEST\(_k\) before the MD associates with DEST\(_k\). Reactive transfer takes place if MD has already associated with DEST\(_k\). In this case, DEST\(_k\) requests the context transfer from SRC\(_k\).

CXTP defines four types of messages: the Context Transfer Activate Request (CTAR) message, the Context Data Transfer (CTD) message, the Context Transfer Request (CT-Request) message and the Context Transfer Cancel (CTC) message. Optionally, the reception of the CTAR or the CTD can be acknowledged by the receiver by a corresponding Context Transfer Activate Acknowledgment (CTAA) or a Context Transfer Data Replay (CTDR) message. The CTC message is sent from the access router in DEST\(_k\) to inform SRC\(_k\) in case the context transfer cannot be completed in a timely fashion.

The predictive and reactive transfer schemes are illustrated in Figure 12.1 and 12.2.

![Figure 12.1: Predictive CXTP](image1)

![Figure 12.2: Reactive CXTP](image2)

If the CTAR message is sent to SRC\(_k\), it includes the IP address of the candidate access router in DEST\(_k\). In case CTAR is sent to DEST\(_k\) it includes the IP address of the previously serving access router in SRC\(_k\). In both cases, the CTAR includes MD’s previous IP address, a sequence number, and information on the context data blocks requested to be transferred. Additionally, CTAR includes an authentication token that is a keyed hash value computed with a key shared between MD and SRC\(_k\) on the concatenation of MD’s previous IP address, the sequence number, and the context data blocks requested to be transferred.\(^1\)

In the predictive case, CTD is sent from SRC\(_k\) to DEST\(_k\) after receipt of a CTAR message from MD. In the reactive case, DEST\(_k\) requests the context transfer from SRC\(_k\) by sending a CT-Request message as soon as it receives a CTAR message from MD.

The CTD message includes the previous IP address of MD, the algorithm, key length, and the key that was used by MD to compute the authentication token, as well as the Context Data Blocks of the actual context. The CTD message is mandated to be sent over

---

\(^1\)How SRC\(_k\) and MD come into possession of the key required to compute the authentication token is out of the scope of CXTP.
12.4 Implementing HEPB Handover Using CARD and CXTP

12.4.1 SRC-controlled HEPB SCT with Key Derivation

Figure 12.3 shows how the HEPB approach can be implemented using CARD and CXTP with predictive context transfer.

Periodically, or triggered by some Layer 2 event, such as a decrease in the reception level of the currently serving network access point, MD scans for candidate destination networks available in its current location. With the help of the CARD protocol, MD discovers the IP addresses and other capabilities of the access routers of surrounding networks by requesting its current access router (CARD-Request). The access router in SRC$_k$ answers with a CARD-Reply that includes the cipher suites $CS_{SRC_{k}}|ssh_{k-1} \cap CS_{DEST_{k}}|ssh_{k-1}$. As all messages exchanged between SRC$_k$ and MD, this message is protected by the temporal keys derived from the pairwise master session key $K_{k-1}$ shared between MD and SRC$_k$. MD selects one of the available access routers and requests context transfer to it from SRC$_k$ by means of a CTAR message. This message is integrity-protected by means of the temporal keys shared between SRC$_k$ and MD, such that R*-8 is met. MD includes its choice $cs_k \in CS_{SRC_{k}}|ssh_{k-1} \cap CS_{DEST_{k}}|ssh_{k-1} \cap CS_{MD}|ssh_{k-1}$ in the context data blocks to be transferred. SRC$_k$ checks whether MD’s choice of $cs_k$ complies with its policy. If this is the case, SRC$_k$ sends the security context $S_k$ as part of the CTD message to DEST$_k$. It is important to note that in the original CXTP, SRC$_k$ includes the key it shares with MD in the CTD message (predictive transfer). Transferring $K_{k-1}$ in CTD, however, contradicts R*-2. In our adapted version of CXTP, SRC$_k$ includes $K_{k-1}$ in CTD. The CTD message is mandated to be sent over an IPsec tunnel between SRC$_k$ and DEST$_k$ such that R*-7 is met. In the meantime, MD disassociates from SRC$_k$ and associates with DEST$_k$. In the association request, MD informs DEST$_k$ of its choice of $cs_k$. DEST$_k$ checks whether $cs_k$ complies with its policy. Before the EAPoL-Handshake starts, MD sends the CTAR to

---

2 Internet Assigned Numbers Authority
DEST<sub>k</sub>. In the original CXTP, MD includes an authorization token in CTAR computed by means of a key shared between MD and SRC<sub>k</sub> and SRC<sub>k</sub> has to transfer the respective key to DEST<sub>k</sub> as part of CTD. Instead, in our adapted version of CXTP, MD computes the authorization token with the derived pairwise master session key <i>K<sub>k</sub></i>. DEST<sub>k</sub> checks whether the sequence numbers included in CTD and CTAR are the same and whether the authentication token included by MD in the CTAR is valid. DEST<sub>k</sub> also checks whether CS<sub>DEST<sub>k</sub></sub><sub>ssh<sub>k</sub>-1</sub> ≠ ∅ and whether <i>T<sub>k</sub> ≤ Tr<sub>DEST<sub>k</sub></sub></i> (R*-1, R*-4). If these checks are successful, DEST<sub>k</sub> and MD perform the EAPoL-Handshake in order to establish fresh data-protection keys <i>T<sub>K</sub></i> from <i>K<sub>k</sub></i>.

As MD chooses the cipher suite to be used after handover, and as SRC<sub>k</sub> sends the context transfer to DEST<sub>k</sub> only if MD’s choice complies with SRC<sub>k</sub>’s policy, and as DEST<sub>k</sub> can drop the connection if MD includes a cipher suite in the association request that does not comply with its policy, the suggested cipher-suite negotiation meets R*-5. As the CARD-Reply message is integrity-protected, the cipher suite negotiation cannot be bid down (R*-6).

Figure 12.4 shows how the HEPB approach can be implemented using CARD and CXTP with reactive context transfer.

As in the predictive case, MD discovers candidate access routers for handover with the help of the CARD protocol. SRC<sub>k</sub> includes CS<sub>SRCh_sshk-1</sub> ∩ CS<sub>DESTk_sshk-1</sub> as one of the capabilities in the CARD-Reply. MD chooses one of the candidate access routers and a cipher suite cs<sub>k</sub> ∈ CS<sub>SRCh_sshk-1</sub> ∩ CS<sub>DESTk_sshk-1</sub> ∩ CS<sub>MD_sshk-1</sub> and includes it in the
12.4 Implementing HEPB Handover Using CARD and CXTP

Association request it sends to DEST
. MD then sends the context transfer request CTAR to DEST including the chosen cipher suite $cs_k$ in the context data and the authentication token, which is computed with the help of $K_k$ and not $K_{k-1}$. DEST requests the context transfer by sending a CT-Request to SRC, including the authorization token and other information received in the CTAR. SRC checks the validity of the authorization token (R*-8) and checks whether the included cipher suite $cs_k$ complies with its policy. Only if these checks are successful, SRC transfers the security context included in CTD to DEST. The CTD message is sent over an IPsec tunnel (R*-7). Upon receipt of CTD, DEST checks whether $CS_{DEST,k} | ssh_{k-1} \neq \emptyset$ and whether $T_k \leq Tr_{DEST,k}$ (R*-1, R*-4). If these checks are successful, DEST extracts $K_k$, and MD and DEST use the EAPoL-Handshake in order to establish a fresh temporal key $TK$ from $K_k$.

The cipher-suite negotiation suggested here is protected against bidding-down attacks (R*-6) in the same way as in the predictive transfer scheme and enforces the policies of SRC$_k$, DEST$_k$, and MD (R*-5).

Note that we assume here that SRC$_k$ has knowledge of the exact and not only a superset of the cipher suites DEST$_k$ allows given the security-suite history $ssh_{k-1}$. Consequently, DEST$_k$ is required to acknowledge any changes of its policies with respect to the allowed cipher suites to SRC$_k$. However, as CARD provides means for SRC$_k$ to inquire DEST$_k$'s current capability, this is a feasible approach.

Figure 12.4: Reactive HEPB-Based Procedure
12.4.2 Changes Required for HN-Controlled and AN-Controlled Handover

CXTP only supports SRC-controlled context transfer upon handover and CARD only supports requesting the currently serving access router for information upon surrounding candidate access routers.

In order to support HN-controlled and AN-controlled handover context transfers, CXTP has to be changed in order to allow CTAR messages sent by MD to be forwarded from SRC to HCN, allow the authentication token to be protected by a key shared between MD and AN, and allow CT-Requests to the HCN rather than SRC. It would be interesting to explore whether these extensions are required by other types of contexts as well.

In order to allow AN or HN to enforce its policies during the negotiation of the cipher suite to be used after handover on AN-controlled and HN-controlled handover, AN (HN) can provide MD with its pre-defined policy sets while MD is associated with AN. Upon handover, MD is then only provided with $CS_{DEST_k}$ as part of the CARD-Reply received from SRC. As in the SRC-controlled case, MD includes its choice of $cs_k$ in the CTAR but protects this message with the pairwise master key shared between MD and AN.

12.4.3 HEPB SCT with Key Agreement

In order to implement HN-controlled HEPB SCT with key agreement (described in Section 5.3) in the WLAN context, MD includes a fresh random number $r$ encrypted with the public handover key of its home network in the CTAR message. SRC forwards the CTAR message to HN. HN partially decrypts $r$ with the share corresponding to DEST_k's share of the secret handover key and includes the partially decrypted $r$ instead of $K_k$ in the security-context transfer in CTD. Upon receipt of the CTD, DEST_k decrypts the partially decrypted value with its share of HN's secret handover key and thus receives $r$. DEST_k and MD use $r$ together with two freshly generated and exchanged random numbers as input TLS-PRF and thus derive the new pairwise master session key. Note that in this case, the security-suite history only consists of the initial authentication and key-agreement protocol.

12.5 Conclusion

In Section 4.5, we already compared our HEPB-based procedure with other SCT-based solutions [177, 162, 186, 74] that are not targeted specifically to the WLAN case.

Duong et al. [55] suggest WLAN-specific security solutions for handover across different providers, combining CARD, CXTP, and Mobile-IP with a scheme to determine the best point in time to initiate a context transfer. This scheme aims to ensure that contexts are in place before MD switches access routers, and at the same time aims to minimize the amount of unnecessary transfers. Note that CARD and CXTP have been finalized only in July 2005; thus, these protocols have not been widely used and analyzed yet.

An important and interesting open topic for future research is to implement and test the HEPB-based approach in a realistic setting. Such an implementation requires implementing
CARD and CXTP, as to the best of the authors knowledge currently no freely available implementation of these protocols exists. The main challenge of implementing the new approach is finding a way to represent the policy expressions of MDs and providers such that the handover decisions as well as the security-mechanism negotiation can be implemented efficiently enough to allow for seamless handover of on-going connections. It would be interesting to investigate whether the policy representation introduced in [179], which allows for efficient reconciliation, could be used for this purpose.

Implementing the new approach will furthermore allow determining a realistic upper bound on the length of context histories that can efficiently be handled dependent on the chosen policy representation. Depending on the resulting maximum length, compressing the context history (see Section 5.1.4) may have to be further investigated.

An interesting side issue in this context is to develop an easy to use interface that allows users and providers to set their policy expressions in a consistent way.

In its current version, CXTP only supports SRC-controlled context transfers and can thus only be used to implement SRC-controlled HEPB SCT. In particular, the HN-controlled SCT with key agreement we suggested in 5.3 cannot easily be implemented in CXTP. We discussed the changes required to CXTP in order to accommodate HN-controlled and DEST-controlled transfer as well, which may be considered as extensions to future versions of the RFC.
Conclusion

In this chapter, we summarize the main results of the thesis, compare them to the most closely related previous work, and point out how our results could be extended in future work.

In the theoretical parts (Part I and II) of this thesis, we have modeled the security challenges imposed on an infrastructure-based wireless access network by inter-provider and inter-system roaming and handover procedures and have developed new security solutions.

We have introduced a formal model for various types of subsequent inter-provider handover procedures. We have identified security-context transfer solutions as the best to-date approach to meet the efficiency requirements imposed by handover procedures. Previous work on inter-provider handover and security ([186, 177, 74, 185, 75, 162]) takes a local view and only models the source and the destination network of a handover, thus ignoring previous handover. As opposed to this, we have taken the anchor network by which a mobile device was originally authenticated and all previously serving networks into account. As a consequence, we have introduced three new control types of subsequent handover procedures that reflect different types of handover agreements between networks.

As stated in [75, 111], the threats arising from security-context transfer across different providers and technologies have been unclear up until now. We have provided a thorough threat analysis of SCT with key derivation and SCT with key agreement, including concrete attack scenarios. Furthermore, we have defined new security requirements and have shown that SCT-based solutions that meet these requirements are secure against the identified attacks. As current solutions, like [162, 176, 177], do not meet our requirements, we have specified history-enriched, policy-based handover procedures for SCT with key derivation [124] and SCT with key agreement. Although our SCT procedures with key derivation meet all but one of our requirements, they still reveal all subsequently used keys to the handover controlling network. For the HN-controlled case, we have solved this problem by presenting a new secret-sharing-based key-agreement method [125]. Future work will explore how the secret-sharing-based key agreement can be extended to inter-provider or inter-system handover controlled by SRC or AN.

The main difference between our procedures and previous work on inter-provider SCT [162, 176, 177] is that our procedures allow mobile devices and networks to express polices with respect to whether or not a handover should take place, dependent on the history of security mechanisms used between a mobile device and any previously serving network.
This protects mobile devices and networks from threats arising from the use of weak security mechanisms before the current handover. Moreover, a mobile device, the destination network, and the handover controlling network can express and enforce policies with respect to the security mechanisms used after handover. This, together with an explicit bidding-down protection of the negotiation of the security mechanisms to use, protects mobile devices and networks from the use of weak security mechanisms after handover.

Furthermore, we have extended the history-enriched, policy-based approach to handover across different technologies. In particular, we have addressed an additional threat arising from differences in the lengths of master keys required by different technologies and have shown how to avoid repeated downgrades of the protection level on each subsequent handover. Our inter-system handover procedures thus far assume that subsequent handover across different technologies are of the same control type. Studying the impact of changing control types is an interesting and challenging topic for future research.

We have modeled and classified roaming authentication and key-agreement protocols for inter-provider and inter-system roaming in a technology independent way and have identified three design goals for roaming security solutions: an ideal roaming authentication and key-agreement protocol should minimize the traffic between the home network and the foreign network, allow for easy handling of changes in roaming profiles and agreements, and keys should be derived where they are used. While the first two design goals are met by previous work on roaming security solutions, none of these solution addresses the last goal.

We have introduced a new approach for public-key-based authentication [121]. This approach has the advantage that, although the home network of a mobile device authorizes the roaming instances of each of its pre-registered users, the key material used to secure the network access is derived by the foreign network. Consequently, no key material has to be transferred over a secure channel between the home network and foreign networks. In all previous work on roaming authentication in which the home network’s authorization is required (e.g., the roaming authentication protocols used in GSM, UMTS, CDMA 2000, all WLAN authentication protocols based on EAP, and the technology-independent solution of Salgarelli et al. [156]), the key material is derived by the home network and needs to be securely transferred to the foreign network. Moreover, our approach solves the problems arising in other public-key-based solutions (e.g., [82] and EAP-TLS used with individual certificates for each foreign network) regarding obtaining and validating chains of certificates on the mobile device. Another suggestion to solve this problem is presented in [29], where the certificate validation is delegated to a trusted server. However, this solution causes additional round-trips to the trusted server. Our new approach uses secret-sharing techniques. Each home network is issued one roaming certificate. The secret key corresponding to this roaming certificate is shared between the home network and its roaming partners. Upon authentication between the foreign network and the mobile device, the home network and the foreign network use a two-party encryption or signature scheme. As a consequence, the home network’s engagement in the authentication is required and at the same time, the identity of the foreign network is guaranteed to the mobile device. Our secret-sharing approach enhances roaming across providers that have pairwise, pre-established roaming
agreements. Extending the approach to other types of roaming agreements, like roaming agreements established on the fly with the help of roaming mediators (see Chapter 2), is another interesting direction for future work.

In Part III of the thesis we have analyzed the roaming and handover procedures within and between GSM and UMTS. The roaming procedures between GSM and UMTS are one of the few examples for standardized inter-system roaming authentication and key-agreement protocols. We have analyzed these roaming procedures and presented a man-in-the-middle attack against the UMTS authentication procedure arising from the inter-operation with GSM [122]. The attack allows an intruder to impersonate a valid GSM base station to a UMTS subscriber, regardless of the fact that an authentication based on UMTS-authentication vectors is used. As a result, the intruder can eavesdrop on all mobile-initiated traffic. Furthermore, we have analyzed all handover procedures standardized within GSM and UMTS, as well as across these technologies [123], and discussed whether they meet our requirements. One of the main results of this analysis is that if the initial authentication of a UMTS subscriber is based on a GSM-authentication vector, a single handover to a GSM network that uses a breakable encryption mechanism, breaks the encryption between a mobile device and a UMTS network after a subsequent handover back to UMTS. This is due to the fact that on handover between GSM and UMTS, only a key-conversion function but no key-derivation function is used. Our history-enriched approach would, nevertheless, allow the UMTS network to detect the use of a suspicious encryption mechanism before handover and consequently refuse the handover. Another result of our analysis is that if an attacker manages to mount a two-sided man-in-the-middle attack against a victim mobile device and a GSM network, he can be handed over to UMTS and thus mount a two-sided man-in-the-middle attack against the UMTS network as well. Our history-enriched policy-based approach would enable the UMTS network to detect this attack and refuse the handover. A thorough analysis of the handover and roaming procedures between UMTS and WLAN that are currently being standardized by 3GPP [12] in our model may reveal weaknesses in the evolving standard.

In Part IV we have applied our new security solutions for roaming and handover to the WLAN case. In this context we have introduced a new roaming authentication protocol EAP-TLS-KS [121] that implements our new approach for IEEE 802.11i-protected WLANs. EAP-TLS-KS exhibits all of the aforementioned advantages of the secret-sharing approach. In addition, compared to regular EAP-TLS usage, EAP-TLS-KS is more efficient in terms of round-trip message exchanges between the home and the foreign network. Moreover, EAP-TLS-KS allows for timely revocation of mobile device certificates and foreign networks’ key-shares. Compromised shares, as well as compromised roaming keys, can easily be revoked and replaced. EAP-TLS-KS is currently being implemented and its performance is evaluated and compared to a regular EAP-TLS implementation in a real-world setting [47]. Moreover, we have shown that our history-enriched, policy-based approach can be used to enhance inter-provider handover between WLANs. We have discussed how the HEPB approach can be integrated with two recently introduced protocols CARD [108] and CXTP [112] to implement mobile-initiated handover across different IP domains. So far, we have
Conclusion

not implemented the HEPB-based approach which leads some open research questions that require further investigation. For once, different methods for representing the policies of users and providers have to be tested with respect to their suitability to enable an efficient security-mechanism negotiation upon handover. In particular, it would be interesting to see, which representation allows for the longest context histories to be processed by HCN and DEST_k efficiently enough to allow for seamless handover. Depending on the results of these tests, methods to compress the context history have to be further evaluated.

Apart from the potential immediate extensions of the results of this thesis already discussed in each paragraph, the thesis motivates future work in the area of security-mechanism negotiation, accounting, and the area of ad hoc or mixed-mode networks:

- The security-mechanism negotiation mechanism Method 5 introduced at the end of Chapter 1 has motivated our current research on privacy-preserving security-policy reconciliation [95].

- With the control types for subsequent handover, we have laid a basis for future research on new accounting models for inter-provider and inter-system handover. Each handover control type naturally supports a certain type of handover agreement, which requires a corresponding accounting scheme. While it seems natural that the home network of a mobile device will be responsible for billing, the clearance between different providers and the extra charges a mobile device will pay for handover support may differ on each control type. For an overview on previous work on this topic, we refer to [64].

- On a broader scale, our threat analysis and the history-enriched, policy-based approach could be extended to infrastructure-based wireless networks that inter-operate with ad hoc networks. In particular, this would require an extension of our roaming and handover model to ad hoc networks. Several additional difficulties arise from handover procedures in ad hoc networks. First, the security model introduced in this thesis cannot easily be adapted to the ad hoc case, as the model is based on centrally stored security-related information. Central network components may not exist in an ad hoc network. Second, in an ad hoc network, the end-point of an IP connection, as well as some, or even all, forwarding intermediate nodes along each route are mobile. Handover procedures of an ongoing connection are consequently not only caused by the sender, but also by each node along the current route.

As we have shown, roaming and handover procedures across different providers and technologies call for new security solutions that enable a secure network access regardless of a user’s location and seamlessly while a user is on the move. While roaming procedures within the same technology have been under discussion since the late 1980s, the security challenges arising from roaming across technologies and inter-provider and inter-system handover across providers have only recently gained larger interest in the research community. While a few years ago querying search machines on “handover and security” would not lead to any results, today’s output can hardly be processed completely. Most current solutions,
however, are technology-specific, and explicit statements on the security requirements of the presented solutions, and the model behind these solutions are often vague. With this thesis, we aim to contribute to a better understanding of the impact of roaming and handover procedures on the security goals of users and providers and present viable solutions to the security challenges imposed by roaming and handover.
Appendix A

Attack Trees

In this appendix, we list the attack trees for the root attack szenarios RAS-1 to RAS-11 on first-order handover procedures. With the help of these trees, we have identified the BAMs, AMs, and attacks described in Section 4.2.

We reuse the notations introduced in Figures 4.1 to 4.6 in Section 4.1. In addition, we introduce a notation to represent several pairs (or larger sets) of AND nodes that alternatively lead to the same subgoal. This notation is illustrated in Figure A.1. In order to achieve A, an attacker can either perform the steps B and C, the steps D and E or the steps F and G.

In some cases, alternative sets of AND nodes make use of the same node. To represent alternative sets of AND nodes we use the notation illustrated in Figure A.2. In order to achieve A, an attacker can either perform the steps B and C or the steps C and D.

---

**Figure A.1: Alternative Pairs of AND Nodes**

**Figure A.2: Alternative Sets of AND Nodes That Make Use of a Common Node**
In some figures, nodes related to recovering $EK_0$ or disabling $em_0$ are dashed. We use this notation to represent nodes that are part of a tree only if signaling traffic in the respective technology is encrypted.

Some of the subtrees used in the attack trees presented here are symmetrical for $EK_i$ and $IK_i$, $i \in \{0,1\}$, or for $em_i$ and $im_i$, $i \in \{0,1\}$. In these cases, we only illustrate one of the subtrees and include a note on what to replace with what in order to obtain the other subtree in the caption of the corresponding figure. In figures that refer to subtrees like these, we add “replace” to the reference if the replacements should be made.

At the end of this appendix, in Figures A.21 to A.26, we identify BAM-1 to BAM-10 and AM-1 to AM-6 within the subtrees of the first-order attack trees. Each of the attack modules AM-1 to AM-6 is highlighted in grey in one of these figures and each of BAM-1 to BAM-10 is identified by the root of the subtree corresponding to the basic attack module.

![Attack Tree for RAS-2](image)

Figure A.3: Attack Tree for RAS-2

![Subtree for Subgoal “Disable $em_0$”](image)

Figure A.4: Subtree for Subgoal “Disable $em_0$”; For “Disable $im_0$” Replace $em_0$ with $im_0$
An attacker gains access to plaintext of data or control traffic exchanged between MD and the real DEST after a handover of MD from HN to DEST took place.

- Recovers plaintext from encrypted traffic intercepted on the air interface between NAP_DEST and MD
- Compromises NAP_DEST and reads plaintext decrypted by NAP_DEST
- Disables encryption between NAP_DEST and MD

- Recovers EK_1
- Partially breaks em_1
- Recovers EK_1 with knowledge of K_1
- Recovers EK_1 without knowledge of K_1
- Totally breaks em_1
- Compromises NAP_DEST and reads EK_1 from memory
- Directly breaks em_1 without manipulating choice of em_1 before

- Recovers K_1 without knowledge of EK_1, IK_1
- Reconstructs ke_1
- Manipulates choice of em_1 to weak encryption mechanism
- Breaks weak em_1 after manipulation
- Manipulates DEST selection to select a DEST that supports weak encryption only
- Manipulates em_1 negotiation to an element of a specific set of weak encryption mechanisms
- Jamms NAPs of other available DESTs
- Amplifies particular NAP_DEST to being strongest available

See Fig. A.6

Figure A.5: Attack Tree for RAS-3
Appendix A. Attack Trees

Bids down em_1 negotiation to 'no encryption'

Bids down em_1 negotiation(1) between MD and HN to 'no encryption'

Bids down em_1 negotiation(3) between MD and HN to 'no encryption'

Bids down em_1 negotiation between HN and DEST to 'no encryption'

Replays recorded negotiation(1) traffic

Recovers EK_0 before negotiation(1) starts

Recovers IK_0 before negotiation(1) starts

Disables encryption between MD and HN

Disables integrity protection between MD and HN

Recover EK_0 before negotiation(1) starts

Replay recorded negotiation(1) traffic

Recover EK_0 without knowledge of EK_1, IK_1

Recovers IK_0 without knowledge of K_1

Disables encryption to an element of a specific set of weak encryption mechanisms

Disables integrity protection to an element of a specific set of weak encryption mechanisms

Recovers EK_0 without knowledge of EK_1, IK_1

Recovers IK_0 without knowledge of EK_1, IK_1

See Fig. A.4

See Fig. A.4

See Fig. A.7, replacement

See Fig. A.7

Figure A.6: Subtree for Subgoal “Manipulate em_1”; For “Manipulate im_1” Replace em_1 with im_1
Appendix A. Attack Trees

Recover EK_0 without knowledge of EK_1, IK_1
Recover EK_0 with knowledge of K_0 without knowledge of EK_1, IK_1
Recover EK_0 without knowledge of K_0
Recover K_0 without knowledge of EK_0 and EK_1, IK_1
Reconstruct ke_0
Reconstructs ke_1
Recover traffic of dynamic ke_1 ke_1 static
Intercepts traffic of dynamic ke_1
Impersonates NAP_DEST to MD on dynamic ke_1

Inverts kd
Recover K_1 without knowledge of EK_0, EK_1, and IK_1 but with knowledge of K_1
Recover K_0 without knowledge of EK_0, EK_1, and IK_1
Recover K_0 without knowledge of EK_0 and EK_1, IK_1
Recover K_0 without knowledge of EK_0, IK_0 and K_1
See Fig. 4.8
See Fig. 4.9
See Fig. 4.10
See Fig. 4.11

Figure A.7: Subtree for Subgoal Combination “EK_0 without K_1 and EK_0 without EK_1, IK_1”; For “IK_0 without K_1 and IK_0 without EK_1, IK_1” replace EK_0 with IK_0.

Figure A.8: Subtree for Subgoal “Reconstruct ke_1”
Figure A.9: Subtree for Subgoal “Recover $K_1$ without $E K_1$ and $I K_1$”; For “Recover $K_1$ without $E K_1$ and $I K_1$ Replace $E K_1$ with $I K_1$.

Figure A.10: Attack Tree for RAS-4
Figure A.11: Subtree for “Impersonate NAP-DEST after a real handover command”
Figure A.12: Subtree for Subgoal “Bids down em₁ to no encryption”; For Subtree “Bids down im₁ to no integrity protection” Replace em₁ with im₁
An attacker gains access to the plaintext of data or control traffic sent by MD by simulating a handover and impersonating NAP_DEST to MD.

- Impersonates NAP_DEST on MD
- NAPDEST association after a handover command is sent to MD
- Sends a faked handover command message to MD
- Changes real or blocks real and inserts new handover command
- Replays previously recorded handover command
- Recovers EK_0 without knowledge of K_1
- Recovers IK_0 without knowledge of K_1
- Disables encryption between MD and HN
- Disables integrity protection between MD and HN

See Fig. A.11
See Fig. A.4, replace
See Fig. A.4, replace
See Fig. A.4

Figure A.13: Attack Tree for RAS-5
An attacker manipulates data or control traffic between MD and the real DEST after handover.

- Recovers IK_1
- Disables the integrity protection between MD and NAP_DEST
- Bids down im_1 negotiation to 'no integrity protection'

See Subtree in Fig. A.5, replace
See Fig. A.12, replace

Figure A.14: Attack Tree for RAS-6
An attacker gains access to HN’s network on behalf of a victim MD

Suppresses handover complete message

Impersonates MD to HN while HN is still waiting for MD to fall back

Recovers EK_0 while HN is still waiting for MD to fall back

Recovers IK_0 while HN is still waiting for MD to fall back

Disables encryption between MD and HN

Disables integrity protection between MD and HN

Replays previously recorded fall back traffic

Recovers EK_0

HN still waits for MD to fall back

Recovers IK_0

See subtree in Fig. 4.7

See subtree in Fig. 4.7, replace

See Fig. A.4

See Fig. A.4

Figure A.15: Attack Tree for RAS-7
An attacker gains access DEST’s network on behalf of a victim MD

Jamms the victim MD such that victim MD and NAP_DEST can not associate

Impersonates the victim MD to NAP_DEST on association

Recover EK_1 before real MD and real NAP_DEST associate

Recover IK_1 before real MD and real NAP_DEST associate

Disables encryption between MD and DEST

Disables integrity protection between MD and DEST

Replays previously recorded association traffic

Compromises NAP_DEST and reads EK_1 from memory before MD and NAP_DEST associate

Compromises NAP_DEST and reads IK_1 from memory before MD and NAP_DEST associate

Recovers K_1 without knowledge of EK_1, IK_1

Recovers K_1

Bids down em_1 negotiation to ‘no encryption’

Bids down im_1 negotiation to ‘no integrity protection’

ke_1 static

See Fig. A.9

See Fig. A.12

See Fig. A.12 (replace em_1 by im_1)

Figure A.16: Attack Tree for RAS-8
Appendix A. Attack Trees

An attacker prevents a legitimate MD to continuously use HN’s or DEST’s service by interfering with the handover procedure.

- **Manipulates em_1,im_1,ke_1 negotiation between MD, HN and DEST fail and thus prevents desirable handover**
- **Sends faked handover command to MD and thus makes MD try to associate to none existent or none prepared NAP_DEST**
- **Jamms MS’s association with DEST and thus disrupts MD or makes MD fall back to HN**
- **Makes HN detect handover reason although there is none**
- **Makes HN select DEST that is not available in the area**
- **Jamms the negotiation traffic between MD and HN**
- **Changes current handover command**
- **Changes negotiation(1) traffic between MD and HN**
- **Replays old handover command previously intercepted**
- **Inserts new faked handover command**
- **Recover EK_0 without knowledge of EK_1, IK_1**
- **Recover IK_0 without knowledge of EK_1, IK_1**
- **Recover EK_0 without knowledge of K_1**
- **Recover IK_0 without knowledge of K_1**

*See Fig. A.17: Attack Tree for RAS-9*
Appendix A. Attack Trees

Figure A.18: Subtree for “False handover detection”

Figure A.19: Attack Tree for RAS-10

Figure A.20: Attack Tree for RAS-11
Recovers $E_k_1$ with knowledge of $K_1$

Recovers $K_1$ without knowledge of $E_k_1$, $I_k_1$

Recovers $K_1$ without knowledge of $E_k_1$, $I_k_1$ and $K_0$

Recovers $K_1$ without knowledge of $E_k_1$, $I_k_1$, but with knowledge of $K_0$

Compute $K_1$ from $K_0$

Recovers $K_0$ without knowledge of $K_1$

Recovers $K_0$ without knowledge of $E_k_0$, $I_k_0$ and $K_1$

Recovers $K_0$ without knowledge of $K_1$ but with knowledge of $E_k_0$, $I_k_0$

Totally breaks $I_k_0$

Recovers $E_k_0$ without knowledge of $K_0$

Recovers $I_k_0$ without knowledge of $K_0$

See Fig. 4.11

See Fig. 4.8

Figure A.21: BAM-1, BAM-2, BAM-3, BAM-7, BAM-8, BAM-9, and AM-1.
Appendix A. Attack Trees

Recovers $E_{K1}$ with knowledge of $K_1$

Reconstructs $K_{e1}$

$BAM-3$

Recovers $K_1$ without knowledge of $E_{K1}$, $I_{K1}$

Recovers $K_1$ without knowledge of $E_{K1}$, $I_{K1}$ and $K_0$

$BAM-9$

Recovers $K_1$ without knowledge of $E_{K1}$, $I_{K1}$, but with knowledge of $K_0$

$Compute K_1 with kd from K_0$

$BAM-2$

Recovers $K_0$ without knowledge of $K_1$

Recovers $K_0$ without knowledge of $E_{K0}$, $I_{K0}$ and $K_1$

$BAM-8$

Recovers $K_0$ without knowledge of $K_1$ but with knowledge of $E_{K0}$, $I_{K0}$

Totally breaks $K_{e0}$

$BAM-1$

Recovers $E_{K0}$ without knowledge of $K_0$

$BAM-7$

Recovers $I_{K0}$ without knowledge of $K_0$

$BAM-7$

See Fig. A.8

See Fig. 4.11

Totally breaks $K_{e0}$

$BAM-1$

Recovers $E_{K0}$ without knowledge of $K_0$

$BAM-7$

Recovers $K_0$ without knowledge of $K_0$ (but only knowledge of $E_{K0}, I_{K0}$)

See Fig. A.8

See Fig. 4.10, replace

Figure A.23: AM-3
Figure A.24: BAM-4, BAM-5, BAM-6, BAM-8, BAM-9, BAM-10, and AM-4.
292 Appendix A. Attack Trees

Recovers $E_K_0$

Recovers $E_K_0$ with knowledge of $K_0$

Recovers $E_K_0$ without knowledge of $K_0$

BAM−7

Recovers $K_0$ without knowledge of $E_K_0$, $I_K_0$

Reconstructs $k_e_0$

BAM−6

BAM−5

Recovers $K_1$ without knowledge of $E_K_0$, $I_K_0$, and $K_0$

Recovers $K_1$ without knowledge of $E_K_0$, $I_K_0$, but with knowledge of $K_1$

Inverts $k_d_0$

BAM−5

Recovers $K_1$ without knowledge of $K_0$

Recovers $K_1$ without knowledge of $E_K_1$, $I_K_1$, and $K_0$

Recovers $K_1$ without knowledge of $E_K_1$, $I_K_1$, but with knowledge of $K_1$

Recovers $E_K_1$ without knowledge of $K_1$

Recovers $I_K_1$ without knowledge of $K_1$

Totally breaks $k_e_1$

BAM−10

BAM−10

BAM−4

See Fig. A.8

See Fig. A.8

See Fig. A.8

See Fig. A.8

See Fig. 4.10

See Fig. 4.10

See Fig. 4.10

See Fig. A.8

Figure A.25: AM-5
Recovers $K_0$ without knowledge of $EK_0$, $IK_0$, and $K_1$
BAM-8

Recovers $K_0$ without knowledge of $EK_0$, $IK_0$
BAM-7

Recovers $K_0$ without knowledge of $EK_0$, $IK_0$, and $K_1$
BAM-6

Reconstructs $ke_0$

Recovers $K_1$ without knowledge of $K_0$
BAM-5

Inverts $kd_0$

Recovers $K_1$ without knowledge of $K_0$, but with knowledge of $K_1$

Recovers $K_1$ without knowledge of $EK_1$, $IK_1$, and $K_0$
BAM-9

Recovers $EK_1$ without knowledge of $K_1$
BAM-10

Recovers $IK_1$ without knowledge of $K_1$
BAM-10

Recovers $EK_0$ without knowledge of $K_0$
BAM-4

Totally breaks $ke_1$

See Fig. A.8

Figure A.26: AM-6
Bibliography


[2] 3GPP. S3-050068: Vulnerabilities and enhancements for GERAN/UTRAN access
security and GSM/UMTS security context, April 2005.

[3] 3GPP. S3-050101: Review of recently published papers on GSM and UMTS security,
February 2005.

[4] 3GPP. S3-050306: Draft LS to GSMA SG on recommendations resulting from a
review of recently published papers on GSM and UMTS security, April 2005.

Project; Technical specification group services and system aspects; Report on the
evaluation of 3GPP standard confidentiality and integrity algorithms, December 2000.

Project; Technical Report: Feasibility study on 3GPP system to wireless local area
network (wlan) interworking, September 2003.

project; 3G Security; Security threats and requirements, December 2001.

[8] 3GPP Technical Specification. 3GPP TS 35.202, V5.0.0, Third Generation Partner-
ship Project; Technical specification group; 3G Security; Specification of the 3GPP
confidentiality and integrity algorithms; Document 2: Kasumi algorithm specification,
June 2002.

[9] 3GPP Technical Specification. 3GPP TS 33.102, V6.3.0, Release 6, Third Generation
Partnership Project; Technical specifications group services and system aspects; 3G

Partnership Project; Technical specifications group services and system aspects; 3G
Security; Specification of the MILENAGE algorithm set: An example algorithm set
of the authentication and key generation functions $f_1, f_1^*, f_2, f_3, f_4, f_5, f_5^*$, document


[92] IEEE. 802.11f - IEEE trial-use recommended practice for multi-vendor access point interoperability via an inter-access point protocol across distributed systems supporting IEEE 802.11 operation, 2003.


[163] 3GPP Technical Specification. 3GPP TS 22.115, V3.2.0, Release 6; third generation partnership project; Technical specification group services and system aspect; Service aspects; Charging and billing, October 1999.


