Introduction

Reconciliation on Ordered Sets
- To enable collaboration of two parties or more parties, it is necessary that they agree on a common rule for this collaboration.
- The process of negotiating this common rule is called a reconciliation protocol.

The Challenge
- Current solutions do not take privacy concerns into account.
- Current solutions do not take preferences of all parties into account.

Example: Scheduling an Appointment Using the iPhone
- Parties: two iPhone users.
- Inputs: a set of possible time slots ordered by preference.
- Goal: determine a time slot that fits best for both parties without revealing time slots or preferences.

Shortcomings of Current Solutions
- At least one party must disclose its complete schedule which leads to privacy concerns.
- New protocols desirable that preserve the privacy of all parties, especially in the multi-party case.

Contributions
- New multi-party reconciliation protocols which meet the privacy requirements of the parties and allow them to find a common rule which optimizes combined preferences.
- Previously, we also designed two-party reconciliation protocols.
- Library for privacy-preserving application for Java, C++, and iPhone platform.

Fair and Privacy-Preserving Multi-Party Reconciliation Protocols

Motivation
- Consider three companies want to schedule a secret project meeting in a privacy-preserving manner.
- Private information or confidential data is given by the possible time slots and associated preferences each party defines (e.g. constrained by the secret project meeting or the Microsoft Free-Day).
- All parties are interested in a common meeting date (gray-bordered time slots).
- All parties are interested in a fair solution.

An intuitive form of fairness is the sum of assigned ranks for each common time slot chosen by all parties.
- Upon completion of the protocol, all parties learn nothing about each other’s ordered sets (time slot and associated preferences), but one common, preference-maximizing set element (optimal common time slot with sum value % green/bordered).

Our Protocols

Protocol Requirements
- Define kind of fairness.
- Find suitable representation and encoding supporting.
- Selection of time slots and associated preferences.
- Evaluation of reconciliation rule.
- Construction of the fairer set according to chosen kind of fairness.
- Calculation of the optimal selection depending on the fairness function.

Privacy-preserving in the same model.

2. Suitable Representation and Encoding
- Multisets chosen as representation form.
- Privacy-preserving intersection (|), union (∪), and element reduction by \( |B| \) defined for multisets.
- Operations performed on encrypted polynomials.
- Multiset \( \rightarrow \) Polynomial: \( N = (n_1, \ldots, n_m) \) represented as \( f(x) = \prod_{i=1}^{m} (x - n_i) \).
- Encoding:
  - Time slots as members of a multiset.
  - Associated preferences encoded in the multiset of the set member.

Performance Evaluation
- Encryption, decryption and homomorphic operations on ciphertexts counted.
- Communication overhead measured in the number of ciphertexts transmitted.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Communication</th>
<th>Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1UMIC</td>
<td>12 ms</td>
<td>2 ms</td>
</tr>
<tr>
<td>2UMIC</td>
<td>12.5 ms</td>
<td>2.5 ms</td>
</tr>
<tr>
<td>3UMIC</td>
<td>13 ms</td>
<td>3 ms</td>
</tr>
<tr>
<td>4UMIC</td>
<td>13.5 ms</td>
<td>3.5 ms</td>
</tr>
</tbody>
</table>

1. Kind of Fairness
- We consider two forms of fairness.
  - Minimum of ranks composition scheme \( g(x) = \min \{rank_{A}(x), \ldots, rank_{N}(x)\} \) for \( x \in B \in \ldots \in B_n \).
  - Sum of ranks composition scheme \( g(x) = rank_{A}(x) + \ldots + rank_{N}(x) \) for \( x \in B \in \ldots \in B_n \).

Solution
- New protocols developed for the functions:
  - Minimum of ranks \( R_{\min}(S_1 \cap \ldots \cap S_n) \).
  - Sum of ranks \( R_{\sum}(S_1 \cup \ldots \cup S_n \cap S'_1 \cap \ldots \cap S'_{n \sim N}) \).

The Algebra class offers an interface to modify polynomials. It includes multiplication with other polynomials, efficient evaluation for plaintext and encrypted coefficients, etc.

Additively Homomorphic Cryptosystem
- Allows transformations of the plaintexts by operations on the ciphertexts.
- Our protocols use the Paillier cryptosystem [3] which, given plaintexts \( m_1, m_2 \) and constant \( c \), has the following properties:
  - \( E(m_1 \cdot m_2) = \lambda_{m_1 + m_2} \).
  - \( E(m_1 + c) = \lambda_{m_1} \).
- These relations allow certain operations on encrypted polynomials used in our multi-party protocols.
  - Sum of encrypted polynomials.
  - Derivative of an encrypted polynomial.
  - Product of unencrypted polynomials and encrypted polynomials.

Polynomial Representation
- Polynomials are well-defined by their coefficients.
- Coefficients are stored in an array according to the order of the terms.
- The Algebra class offers an interface to modify polynomials. It includes multiplication with other polynomials, efficient evaluation for plaintext and encrypted coefficients, etc.

Implementation
- To verify the theoretically determined time-complexity as well as the correctness of our protocols, we are developing a library for privacy-preserving applications. The library consists of a Java, C++, and iPhone implementation. The protocols and all required components were implemented from scratch.

Building Blocks of the Java PPA Library
- Add-on for the iPhone App.
- Current implementation is based on Bluetooth allowing the use of the tool when both parties are in the same physical location.

Future Work
- Design protocols to be privacy-preserving in the malicious model.
- Investigate into further suitable preference composition schemes.
- Protocol optimizations.
- Extend appoint to the multi-party case.

References