# A Secure Two-Party Bartering Protocol
## Using Privacy-Preserving Interval Operations*

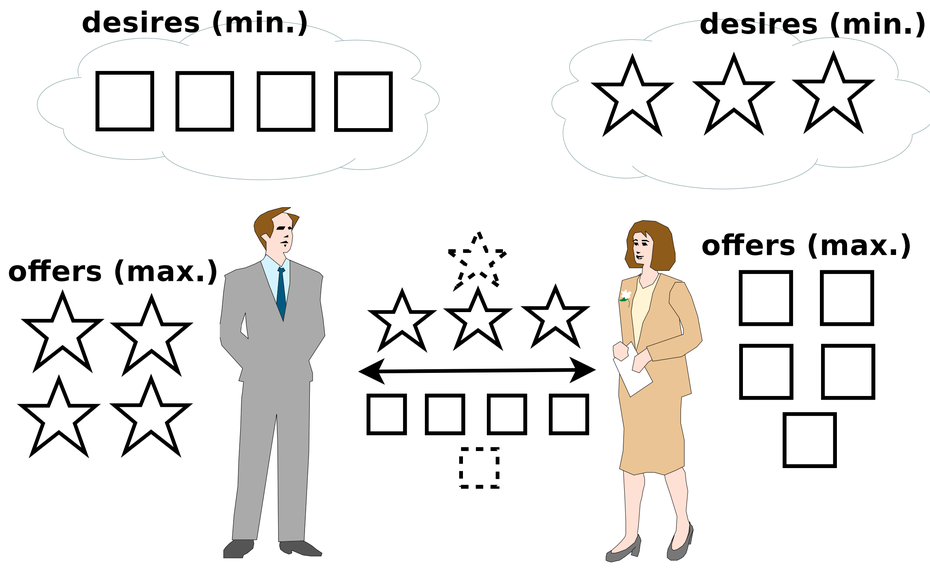
Fabian Förg[1], Daniel Mayer[1], Susanne Wetzel[1], Stefan Wüller[2], Ulrike Meyer[2]

[1] Stevens Institute of Technology, Hoboken, NJ, USA [2] RWTH Aachen University, Aachen, Germany

* Published at the Annual Conference on Privacy, Security and Trust (PST) 2014

## Bartering

Practice of trading goods or services in exchange for other goods or services, rather than for money.




**Shortcomings of Previous Work**:

- Manual trading process: meet face-to-face or exchange messages online
- Reveal personal preferences: what to trade and which quantities

**Goals of Our Work**:

- **Automation**: trading process
- **Privacy-Preserving**: keep personal preferences private
- **Unbiasedness**: honor preferences without favoring a particular party

## Formalization

| | | Party $P_A$ | Party $P_B$ |
|---|---|---|---|
| commodities | desired | $c_d^{(A)}$ | $c_d^{(B)}$ |
| | offered | $c_o^{(A)}$ | $c_o^{(B)}$ |
| quantity ranges | desired | $Q_d^{(A)} = [\underline{q}_d^{(A)}, \infty)$ | $Q_d^{(B)} = [\underline{q}_d^{(B)}, \infty)$ |
| | offered | $Q_o^{(A)} = (0, \overline{q}_o^{(A)}]$ | $Q_o^{(B)} = (0, \overline{q}_o^{(B)}]$ |

Trade condition:

- Desired and offered commodities mutually match **and**
- Desired and offered quantity ranges mutually overlap

Selection of quantity to trade:

- Both parties are willing to trade quantities in overlap
- Selection of quantity to trade uniform at random out of overlap does not favor party (unbiasedness)

## Bartering Algorithm (Not Privacy-Preserving)

- Public set of commodities $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$
- Representation of $c_i$ $(1 \leq i \leq |\mathcal{C}|)$: $(0, \ldots, 0, \underbrace{1}_{i\text{-th component}}, 0, \ldots, 0) \in \{0,1\}^{|\mathcal{C}|}$

1 $s_1 := c_d^{(A)} \times c_o^{(B)}$ /* Commodities match? */
2 $s_2 := c_d^{(B)} \times c_o^{(A)}$
3 $c_1 := \underline{q}_d^{(A)} \overset{?}{\leq} \overline{q}_o^{(B)}$ /* Quantities compatible?
4 $c_2 := \underline{q}_d^{(B)} \overset{?}{\leq} \overline{q}_o^{(A)}$
5 **if** $s_1 \wedge s_2 \wedge c_1 \wedge c_2$ /* *Trade exists?* */ **then**
6 $\quad t_{c_o^{(B)}} \leftarrow_\$ Q_d^{(A)} \cap Q_o^{(B)}$ /* Select quantity to trade for $c_o^{(B)}$ */
7 $\quad t_{c_o^{(A)}} \leftarrow_\$ Q_d^{(B)} \cap Q_o^{(A)}$ /* Select quantity to trade for $c_o^{(A)}$ */
8 **else**
9 $\quad t_{c_o^{(B)}} := 0$
10 $\quad t_{c_o^{(A)}} := 0$
11 **return** $(t_{c_o^{(B)}}, t_{c_o^{(A)}})$

## Cryptographic Background

- $(\mathbf{t}, \mathbf{n})$ **threshold cryptosystem**: at least $t$ of $n$ parties are necessary to jointly decrypt a ciphertext, while any single party is able to encrypt
- **Semantic security**: given only a ciphertext, it is infeasible to learn anything about its plaintext
- **Computations on ciphertexts**: an additive homomorphic cryptosystem allows to efficiently compute $E(m_1) +_h E(m_2) = E(m_1 + m_2)$ and to re-randomize a ciphertext $c$ by computing $\mathcal{R}(c) := c +_h E(0)$
- **Secure multi-party computation**: parties compute the output correctly, but must not learn more than what they can deduce from their own input and the output
- **Semi-honest model**: parties follow the protocol, but attempt to learn more than what they can deduce from their own input and the output

## Intuition of Privacy-Preserving Protocol

- Input: commodities and quantities
- Output: quantities to trade

Output reveals whether or not trade condition holds
→ Sufficient to determine ciphertext of trade condition and decrypt
→ Two steps: (1) Determine ciphertext for trade condition (2) Select quantity to trade

1. Determine ciphertext for trade condition, while *keeping commodities and quantities private*
   - Privacy-preserving subprotocols exist for:
     - scalar product: commodities match? [1]
     - comparison with shared output ($\mathcal{F}_{\text{SC-SO}_{\text{LT}}}$): quantities compatible? [2]
   - Intertwine subprotocol steps: *oblivious combination* of intermediate ciphertext and input using:
     - AND (combine clauses in trade condition)
     - XOR (combine shared output of comparison protocol)

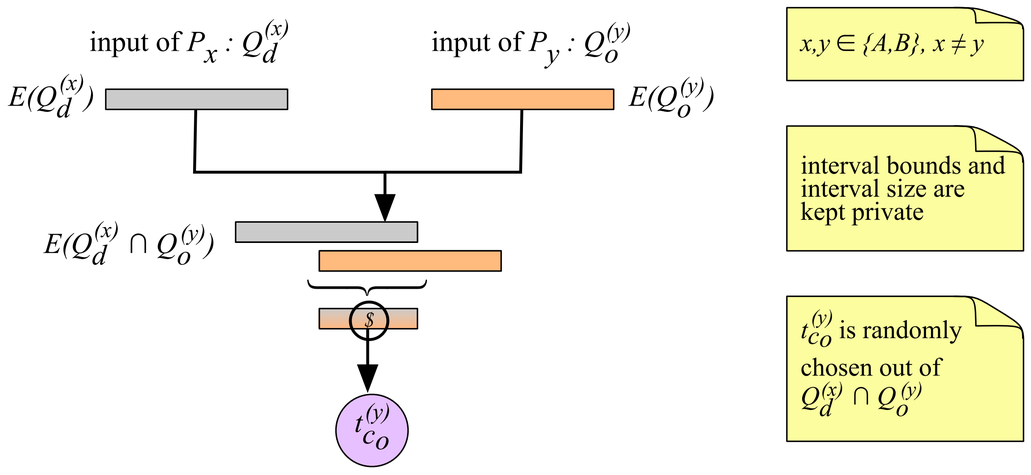
$P_A : E(a)$ AND $P_B : b \in \{0,1\}$

$c := E(a)$ $\xrightarrow{c}$ $E(a \wedge b) = \begin{cases} E(0) & \text{if } b = 0 \\ \mathcal{R}(c) & \text{if } b = 1 \end{cases}$

$P_A : E(a), E(\neg a)$ XOR $P_B : b \in \{0,1\}$

$c_1 := E(a)$, $c_2 := E(\neg a)$ $\xrightarrow{c_1, c_2}$ $E(a \oplus b) = \begin{cases} \mathcal{R}(c_1) & \text{if } b = 0 \\ \mathcal{R}(c_2) & \text{if } b = 1 \end{cases}$

2. Select quantity to trade, while *keeping quantities private* ($\mathcal{F}_{\text{RSI}}$)




## Theoretical Performance Analysis

| Type of Operation | $\mathcal{F}_{\text{barter}}$ |
|---|---|
| # Homomorphic Additions | 3 |
| # Scalar Multiplications | 0 |
| # Encryptions | $2 \cdot \lvert\mathcal{C}\rvert + 2$ |
| # Decryptions | 1 |
| # Messages | 4 |
| Payload in # Ciphertexts | $2 \cdot \lvert\mathcal{C}\rvert + 4$ |
| # Executions of $\mathcal{F}_{\text{SC-SO}_{\text{LT}}}$ | 2 |
| # Executions of $\mathcal{F}_{\text{RSI}}$ | 0/2 (best/worst case) |

- $\mathcal{F}_{\text{RSI}}$ has computation complexity of $\Theta(n^2)$ where $n$ is the upper bound of the unsigned integers to compare
- $\mathcal{F}_{\text{SC-SO}_{\text{LT}}}$ from [2] has computation complexity of $\mathcal{O}(log(n) + s)$ where $s$ is the security parameter of the underlying cryptosystem

## Future Work

- Baskets of commodities (with respective quantities)
- Multiple parties
- Stronger adversary models

## References

[1] Zhigiang Yang, Rebecca N. Wright, and Hiranmayee Subramaniam. Experimental Analysis of a Privacy-Preserving Scalar Product Protocol. *International Journal of Computer Science & Engineering*,21,2006.

[2] Ahmet Erhan Nergiz, Mehmet Ercan Nergiz, Thomas Pedersen, and Chris Clifton. Practical and Secure Integer Comparison and Interval Check. *2010 IEEE Second International Conference on Social Computing*, 2010.