Towards Privacy-Preserving Multi-party Bartering

Stefan Wüller[1(✉)], Ulrike Meyer[1], and Susanne Wetzel[2]

[1] RWTH Aachen University, Aachen, Germany
{wueller,meyer}@itsec.rwth-aachen.de
[2] Stevens Institute of Technology, Hoboken, NJ, USA
swetzel@stevens.edu

Abstract. Both B2B bartering as well as bartering between individuals is increasingly facilitated through online platforms. However, typically these platforms lack automation and tend to neglect the privacy of their users by leaking crucial information about trades. It is in this context that we devise the first privacy-preserving protocol for automatically determining an actual trade between multiple parties without involving a trusted third party.

1 Introduction

The Encyclopedia Britannica defines *bartering* as "the direct exchange of goods or services—without an intervening medium of exchange or money—either according to established rates of exchange or by bargaining". Bartering is considered to be the oldest form of trading and has been practiced since the early days of humanity. In this traditional form, bartering typically requires a party to find a single trade partner that offers what the party demands and at the same time demands what the party offers. Alternatively, it may try to find a larger trade cycle in which more than two parties will exchange their goods or services in a cyclic fashion. While the former may not even exist, the latter is difficult if not even impossible to find. In any case, the offer and demand of each party needs to be satisfied simultaneously.

The introduction of currencies resolved these issues to some extent. In particular, it allows to decouple the search for a trade partner that satisfies a partys demand from the search for a trade partner that demands what that party offers. In addition, traditional trading with (cash) currencies guarantees that each party only learns how much of what it is selling to whom and how much of what it is buying from whom but nothing about what their trade partners do in return. Also there is no bank or any other trusted third party directly involved in the trading, observing who buys what from whom. The importance of these privacy guarantees offered by cash currencies are widely recognized and have led to the introduction of many successful digital counterparts (e.g., Bitcoin [11]).

Despite the benefits of using money as a mediator in trading, bartering has become popular again in recent years. This is due, among other reasons, to the

M. Brenner et al. (Eds.): FC 2017 Workshops 2017, LNCS 10323, pp. 19–34, 2017.
https://doi.org/10.1007/978-3-319-70278-0_2

fact that online bartering platforms greatly facilitate the cumbersome search for trade partners (e.g., U-Exchange, BarterQuest, or TradeYa). However, these platforms typically disclose what (and how much) parties seek or offer at least to the operator of the platform and typically also to other parties even if a trade between these parties is not possible. Thus the privacy guarantees offered by traditional bartering (i.e., a party merely learns what it gets and what it gives away and there is no third party observing the transactions) are lost.

The goal of our work is to follow suit with digital cash and enable electronic bartering with privacy guarantees equivalent to the guarantees provided by traditional bartering or trading using (cash) currencies. In our bartering process, each party specifies a quote defining its offered and desired commodity and the corresponding quantity ranges. A party keeps its quote private at all times from all other parties. Upon completion of the privacy-preserving bartering process, each party learns nothing but its direct trade partners as well as the commodities and quantities to be sent and received. Thus, for a given set of parties and their quotes, our bartering process privately determines an actual trade which includes the actual trade constellation of the parties (i.e., which party trades with which other party) as well as the actual commodities and quantities to be traded. The actual trade can be selected based on different selection strategies including the maximization of the number of parties able to trade. At the core of our bartering process (designed as a secure multi-party protocol based on homomorphic threshold encryption) is a novel protocol that privately determines the actual trade constellation. This protocol makes use of a novel privacy-preserving mapping operation that is based on the uniqueness of prime factorization, which is of independent interest beyond the context of electronic bartering.

Obviously, given their local view of the actual trade constellation, the parties can negotiate the quantities at which the commodities are to be exchanged outside of the bartering process described above. Yet, it is important to recognize that in practice this requires one of the parties to first state its intentions. In order to compensate for such a disadvantage, a party may elect to lie about the range it is willing to accept. As a first step to mitigating this problem, we enable to negotiate the actual quantities in an automatic and unbiased fashion by randomly sampling out of a private interval (defined by the private limits of the parties). As such, this approach motivates the parties to privately specify their true negotiation ranges.

2 Related Work

For the two-party case, secure multi-party computation (SMPC) protocols for privacy-preserving bartering have been proposed, e.g., in [5,7]. While these two-party protocols can obviously be used to find pairwise trades in the multi-party setting with more than two parties as well, they cannot be used to determine trade cycles between more than two parties. The particular challenge of finding such cycles in a privacy-preserving way in the SMPC setting has already been recognized in [6] but has not been addressed so far.

To the best of our knowledge, there is only one approach to privacy-preserving multi-party bartering that has been proposed in the past [9]: Kannan et al. introduce a protocol where each party holds an indivisible commodity from a publicly known finite set of commodities as well as a totally ordered preference list over all commodities in the set. Their goal is then to determine an actual trade between multiple parties such that the computed commodity allocation is pareto optimal while the input of each party (commodity and preference list) is kept private. Specifically, the protocol protects the parties' input under the notion of *marginal differential privacy* [9] which is a relaxation of *differential privacy* [4]. In contrast to differential privacy, marginal differential privacy is restricted to an adversary that has access to the protocol output of only one single party which corresponds to the assumption that there are no colluding parties participating in the protocol which try to subvert the privacy of another party. The substantial difference between the approach from [9] and our approach is that the former one focalizes on the privacy of the parties' input after the functionality is computed while the major goal of our approach is to provide privacy during the computation of an actual trade. Further differences to our work are that the protocol from [9] requires a trusted third party in order to determine an actual trade and that they use a weaker privacy notion that assumes non-colluding parties. In our approach, an actual trade is computed without the help of a trusted third party and we allow that all but one colluding parties may be controlled by an adversary. In addition, our approach supports divisible commodities.

In contrast to e-commerce (and auctions), bartering transactions are not necessarily reduced to money which allows for a richer structure of exchanges [10]: A trade takes place if the involved parties are satisfied w.r.t. the specification of their offered and desired commodities and the corresponding quantities. If the commodities first have to be converted into money (as it is the case for e-commerce and auctions), the prices of the commodities have to be individually determined. Consequently, a party desiring a commodity which is more expensive than its offered commodity is not able to barter, although a trade could have taken place if the commodities were traded directly [10]. Thus, privacy-preserving protocols for e-commerce scenarios (e.g., [1]) or auctions (e.g., [12]) can not directly be applied to implement privacy-preserving bartering.

3 Preliminaries

By $a \leftarrow_{\$} A$ we indicate that a is drawn uniformly at random from A. $\mathbb{N}_u := \{1, ..., u\}$ refers to the set of natural numbers less than or equal to $u \in \mathbb{N}$. The set of all prime numbers within an integer interval I is referred to as $\mathbf{P}_I$. We denote the index set of all parties P_i participating in a multi-party protocol as $\mathscr{P} := \{1, \ldots, \iota\}$ where $i \in \mathscr{P}$. Furthermore, λ denotes the empty string.

3.1 Threshold Paillier

Our design approach assumes an additively homomorphic cryptosystem which is semantically secure against chosen-plaintext attacks and provides a (τ, ι) thresh-

old variant, i.e., the decryption key is distributed amongst ι parties such that at least $\tau \leq \iota$ parties have to collaborate in order to decrypt a ciphertext.

In the following, we summarize the (τ, ι) threshold variant of the Paillier cryptosystem [13] from [3] along with the Paillier-related notation used throughout the paper.

The public key corresponds to an RSA modulus $N = p \cdot q$ of bit length k, where p, q are safe primes (i.e., there are prime numbers p' and q' such that $p = 2p' + 1$ and $q = 2q' + 1$) and k refers to the security parameter. The private key $d \in \mathbb{Z}_{p'q'N^s}$ with $s > 0$, $s \in \mathbb{N}$ satisfying $d = 0 \ mod \ p'q'$ and $d = 1 \ mod \ N^s$ is polynomially shared between $P_1, \ldots, P_\iota$ such that at least τ parties have to cooperate for decryption. The encryption of a message m in the *plaintext space* $\mathbb{P} := \mathbb{Z}_{N^s}$ is computed as $c = E(m) := (N+1)^m r^{N^s} \ mod \ N^{s+1}$ where $r \leftarrow_\$ \mathbb{Z}^*_{N^{s+1}}$ and c is an element in the *ciphertext space* $\mathbb{C} := \mathbb{Z}^*_{N^{s+1}}$. Throughout the paper we assume $s = 1$. We have that the plaintext space $\mathbb{P}$ forms the additive group $(\mathbb{Z}_N, +)$, and the ciphertext space $\mathbb{C}$ forms the multiplicative group $(\mathbb{Z}^*_{N^2}, \cdot)$. For further details we refer to [3].

Let $m, m_1, m_2 \in \mathbb{P}$ and $\kappa \in \mathbb{N} \setminus \{0\}$. The Paillier ($(\tau, \iota)$ threshold) cryptosystem provides for *homomorphic addition*

$$E(m_1) +_h E(m_2) := E(m_1) \cdot E(m_2) = E(m_1 + m_2)$$

and *homomorphic scalar multiplication*

$$E(m) \times_h \kappa := \underbrace{E(m) \cdot E(m) \cdots E(m)}_{\kappa \text{ times}} = E(\kappa \cdot m).$$

A ciphertext $E(m)$ can be *randomized* (or *re-randomized*) by homomorphically adding a fresh encryption of zero. For the remaining sections $\mathbb{P}$, $\mathbb{C}$, $E(\cdot)$, and $D(\cdot)$ refer to the plaintext space, the ciphertext space, the encryption function, and the decryption function of (τ, ι) threshold Paillier, respectively. Note that for convenience, we omit the keys from the notation.

3.2 Secure Multi-party Computation

In order to define security comprising privacy and correctness, we have to specify the capabilities of an adversary under whose presence a protocol has to be secure. We prove our protocols to be secure in the semi-honest model. A semi-honest adversary controls a set of corrupted parties which correctly follow the protocol specification with the exception that each corrupted party keeps record of all data it generates itself and all messages it receives from other parties.

We assume that the parties communicate over authentic channels, i.e., the transferred data is resistant to tampering but can be wiretapped.

Let $\widehat{X} := (X_1, \ldots, X_\iota)$ and let $\mathcal{F} : (\{0,1\}^*)^\iota \to (\{0,1\}^*)^\iota, \widehat{X} \mapsto (\mathcal{F}_1(\widehat{X}), ..., \mathcal{F}_\iota(\widehat{X}))$ be a multi-party ($|\mathscr{P}| = \iota \geq 2$) functionality computable in polynomial time where P_i provides input X_i and obtains output $\mathcal{F}_i(\widehat{X})$ ($i \in \mathscr{P}$). Let π be an ι-party protocol for computing functionality $\mathcal{F}$. We write $I_C :=$

$\{i_1, ..., i_\kappa\} \subset \mathscr{P}$ for the index set of $1 \leq \kappa < \iota$ corrupted parties controlled by the adversary. The view of P_i during an execution of π on input $\widehat{X}$ and security parameter s is denoted as $\text{VIEW}_i^\pi(s, \widehat{X}) := (s, X_i, \mathring{r}_i, m_{i,1}, ..., m_{i,n})$, where $\mathring{r}_i$ represents P_i's internal random tape and $m_{i,j}$ represents the j-th message P_i received during a protocol execution of π. We write $\text{OUTPUT}^\pi(s, \widehat{X}) := (\text{OUTPUT}_1^\pi(s, \widehat{X}), \ldots, \text{OUTPUT}_\iota^\pi(s, \widehat{X}))$ in order to refer to the output of protocol π on input $\widehat{X}$ and security parameter s. Let $\widehat{X}_{I_C}$, $\mathcal{F}_{I_C}(\widehat{X})$, and $\text{VIEW}_{I_C}^\pi(\widehat{X})$ denote the κ-tuples $(X_{i_1}, ..., X_{i_\kappa})$, $(\mathcal{F}_{i_1}(\widehat{X}), ..., \mathcal{F}_{i_\kappa}(\widehat{X}))$, and $(I_C, \text{VIEW}_{i_1}^\pi(\widehat{X}), ..., \text{VIEW}_{i_\kappa}^\pi(\widehat{X}))$, respectively.

Definition 1 (Security: Semi-Honest Model, Multi-Party Setting [8]). *π securely computes $\mathcal{F}$ if there exists a probabilistic polynomial time algorithm $\mathcal{S}$ such that for every I_C it holds that $\{(\mathcal{S}(1^s, I_C, \widehat{X}_{I_C}, \mathcal{F}_{I_C}(\widehat{X})), \mathcal{F}(\widehat{X}))\}_{\widehat{X},s}$ and $\{(VIEW_{I_C}^\pi(\widehat{X}, s), OUTPUT^\pi(s, \widehat{X}))\}_{\widehat{X},s}$ are computational indistinguishable.*

For convenience, we omit s from the remaining considerations. We call $\mathcal{S}$ a *simulator* and enclose the values it *simulates* by square brackets $\langle \cdot \rangle$ in order to distinguish between simulated values and those occurring during a protocol run.

In order to facilitate the security proof of a protocol π implementing functionality $\mathcal{F}$ where π consists of a finite set of sub-protocols $\rho_1, \ldots, \rho_n$ securely computing functionalities $\mathcal{G}_1, \ldots, \mathcal{G}_n$ in the semi-honest model, we can apply the *Modular Composition Theorem* [2] which states that if π' securely computes $\mathcal{F}$ in the semi-honest model where the sub-protocol calls of π are replaced by calls to a trusted third party computing $\mathcal{G}_1, \ldots, \mathcal{G}_n$, then π securely computes $\mathcal{F}$ in the semi-honest model.

To prove our protocols to be secure in the semi-honest model, we first prove that $\{\mathcal{F}(\widehat{X})\}_{\widehat{X}} \stackrel{c}{\equiv} \{OUTPUT^\pi(\widehat{X})\}_{\widehat{X}}$. This step is referred to as *Correct Output Distribution* (COD). Second, we prove that $\text{VIEW}_{I_C}^\pi$ can be simulated under consideration of the given inputs and outputs of all corrupted parties such that $\text{VIEW}_{I_C}^\pi$ and the corresponding simulated view are computationally indistinguishable, referred to as *Correct View Distribution* (CVD).

To refer to a concrete functionality or protocol, we use the templates $\mathcal{F}_{name}^{[affix]}$ and $\pi_{name}^{[affix]}$ where protocol $\pi_{name}^{[affix]}$ is an implementation of functionality $\mathcal{F}_{name}^{[affix]}$ with *name* and *affix* describing the functionality to be computed where the use of *affix* is optional. For convenience, we omit *name* and *affix* for the case that the target functionality and protocol is clear from the context. Furthermore, we write $\mathcal{F}(X_1, \ldots, X_\iota, X)$ to denote that X is a public input that is known by all parties. $(o) \leftarrow \mathcal{F}(X)$ indicates that all parties have common input X and common output o.

4 Overview

4.1 Bartering Related Terminology

For a set of parties, a trade generically indicates which party receives (or sends) which quantity of which commodity from (or to) which other party. In this

Table 1. Bartering related acronyms used throughout the paper.

$TPT(S)$	Trade partner tuple (Set)	Definition 2 (below Definition 10)
$TPC(S)$	Trade partner constellation (Set)	Definition 3 (below Definition 4)
$PTPC(S)$	Potential trade partner constellation (Set)	Definition 4 (below Definition 4)
$ATPC$	Actual trade partner constellation	Definition 5
AT	Actual trade	Definition 6

paper, we focus on so-called (1:1) *trades* with one offered and one desired commodity for each party. In such a trade, each party receives some quantity of its desired commodity from at most one party and sends some quantity of its offered commodity to at most one other party.

More specifically, we consider a set of ι parties $\{P_i | i \in \mathscr{P}\}$ with $\mathscr{P} := \mathbb{N}_\iota$ and a publicly known finite set $\mathscr{C} := \{c_1, \ldots, c_n\}$ of divisible commodities. Each party P_i specifies exactly one *quote* $\mathbf{q}^{(i)} := (\mathbf{o}^{(i)}, \mathbf{d}^{(i)})$ where $\mathbf{o}^{(i)}$ and $\mathbf{d}^{(i)}$ is P_i's *offer* and *demand*, respectively. We model $\mathbf{o}^{(i)}$ as a 3-tuple $\mathbf{o}^{(i)} := (c_o^{(i)}, \underline{q}_o^{(i)}, \overline{q}_o^{(i)})$ where $c_o^{(i)} \in \mathscr{C}$ specifies the commodity offered by P_i and $\underline{q}_o^{(i)} \in \mathbb{N}\backslash\{0\}$ ($\overline{q}_o^{(i)} \in \mathbb{N}\backslash\{0\}$) denotes the minimum (maximum) quantity of $c_o^{(i)}$ offered. Similarly, we model $\mathbf{d}^{(i)} := (c_d^{(i)}, \underline{q}_d^{(i)}, \overline{q}_d^{(i)})$ with $c_d^{(i)} \in \mathscr{C}$ and $\underline{q}_d^{(i)}, \overline{q}_d^{(i)} \in \mathbb{N}\backslash\{0\}$. With $\mathbf{q}^{(i)}$ a party P_i indicates that it is *satisfied* with a trade if it receives at least $\underline{q}_d^{(i)}$ and at most $\overline{q}_d^{(i)}$ units of commodity $c_d^{(i)}$ and sends at least $\underline{q}_o^{(i)}$ and at most $\overline{q}_o^{(i)}$ units of $c_o^{(i)}$. For convenience, we assume that $\underline{q}_o^{(i)} = 1$ and $\overline{q}_d^{(i)} = \infty$. The *quantity ranges* of the offered and desired commodities of a party P_i ($i \in \mathscr{P}$) are thus defined as $Q_o^{(i)} := [1, \overline{q}_o^{(i)}]$ and $Q_d^{(i)} := [\underline{q}_d^{(i)}, \infty]$. We write $q_{c_o^{(i)}}^{(i,i')}$ in order to indicate at which quantity $P_{i'}$ will receive commodity $c_o^{(i)}$ from P_i ($i, i' \in \mathscr{P}$).

We introduce the following bartering related terms which are summarized in Table 1 and illustrated in Fig. 1:

Definition 2 (Trade Partner Tuple). *A* trade partner tuple $TPT^{(i)} := (x^{(i)}, y^{(i)})$ *for* P_i ($i \in \mathscr{P}$) *with* $x^{(i)}, y^{(i)} \in \mathscr{P}\backslash\{i\}$ *is a 2-tuple which specifies the indices of the trade partners* $P_{x^{(i)}}$ *and* $P_{y^{(i)}}$ *of* P_i*:* $P_{x^{(i)}}$ *is the offerer of party* P_i*, i.e.,* P_i *receives some quantity of some commodity from* $P_{x^{(i)}}$*, while* $P_{y^{(i)}}$ *is the demander of* P_i*, i.e.,* P_i *has to send some quantity of some commodity to* $P_{y^{(i)}}$*. If a party* P_i *neither sends nor receives any commodity in a trade, i.e., it does not participate, we write* $TPT^{(i)} = (0, 0)$.

Definition 3 (Trade Partner Constellation). *A trade partner constellation* $TPC := (TPT^{(1)}, TPT^{(2)}, ..., TPT^{(\iota)})$ *is an* ι*-tuple which specifies exactly one trade partner tuple for each* P_i ($i \in \mathscr{P}$) *and has the following property: for each trade partner tuple* $TPT^{(i)} = (x^{(i)}, y^{(i)})$ *it either holds that* $x^{(i)} = y^{(i)}$ *or it holds that there exist exactly two distinct entries* $TPT^{(i')}$ *and* $TPT^{(i'')}$ *with* $i \neq i', i''$ *such that* $TPT^{(i')} = (y^{(i)}, y^{(i')})$ *and* $TPT^{(i'')} = (x^{(i'')}, x^{(i)})$.

Definition 3 ensures that each party that participates as offerer (demander) in some TPT of a TPC also participates as demander (offerer) either in the same or in exactly one other TPT of the TPC.

For a fixed context of quotes $\mathbf{Q}: = \{\mathbf{q}^{(1)}, ..., \mathbf{q}^{(\iota)}\}$ with $\mathbf{q}^{(i)} = ((c_o^{(i)}, \underline{q}_o^{(i)}, \overline{q}_o^{(i)}), (c_d^{(i)}, \underline{q}_d^{(i)}, \overline{q}_d^{(i)}))$, a TPC is transformed into a *trade partner constellation formula*, written $\varphi \overset{\mathbf{Q}}{\sim} TPC$, such that:

$$\varphi: = \bigwedge_{\substack{i=1 \\ (x^{(i)}, y^{(i)}) \neq (0,0)}}^{\iota} \mathcal{C}(\mathbf{q}^{(i)}, \mathbf{q}^{(x^{(i)})}) \wedge \mathcal{R}(\mathbf{q}^{(i)}, \mathbf{q}^{(x^{(i)})}) \tag{1}$$

with

$$\mathcal{C}(\mathbf{q}^{(a)}, \mathbf{q}^{(b)}): = \begin{cases} 1 & \text{if } (\mathbf{q}^{(a)}, \mathbf{q}^{(b)}) \in C \\ 0 & \text{otherwise} \end{cases}, \mathcal{R}(\mathbf{q}^{(a)}, \mathbf{q}^{(b)}): = \begin{cases} 1 & \text{if } (\mathbf{q}^{(a)}, \mathbf{q}^{(b)}) \in R \\ 0 & \text{otherwise} \end{cases}$$

where

$$C: = \{(\mathbf{q}^{(a)}, \mathbf{q}^{(b)}) | c_d^{(a)} = c_o^{(b)}\},\ R: = \{(\mathbf{q}^{(a)}, \mathbf{q}^{(b)}) | \underline{q}_d^{(a)} \leq \overline{q}_o^{(b)}\}.$$

Evaluating φ (for a given context of quotes) denoted as $[\![\varphi]\!] \in \{0, 1\}$ allows one to check whether or not there is a trade which all parties P_i (with $TPT^{(i)} \neq (0, 0)$) in the corresponding trade partner constellation are satisfied with. The trade partner constellations for which this holds for a given context of quotes $\mathbf{Q}$ are referred to as *potential trade partner constellations*:

Definition 4 (Potential Trade Partner Constellation). *For a context of quotes* $\mathbf{Q}$*, a trade partner constellation* TPC *is a potential trade partner constellation* ($PTPC$)*, iff* $\varphi \overset{\mathbf{Q}}{\sim} TPC$ *and* $[\![\varphi]\!] = 1$.

We write $TPCS: = \{TPC_1, \ldots, TPC_t\}$ for a set of trade partner constellations. Given $TPCS$ and $\mathbf{Q}$, the set of potential trade partner constellations is denoted as $PTPCS$. Furthermore, given $TPCS$ and $\mathbf{Q}$, we define $\Phi: = \{\varphi_j | \varphi_j \overset{\mathbf{Q}}{\sim} TPC_j, j \in \mathbb{N}_{|TPCS|}\}$ and $\Phi_{\text{sat}}: = \{\varphi_j | \varphi_j \in \Phi, [\![\varphi_j]\!] = 1\} \subseteq \Phi$.

Definition 5 (Actual Trade Partner Constellation). *An actual trade partner constellation* $ATPC$ *is a specific* $PTPC$ *drawn from* $PTPCS$ *based on a specified selection strategy.*

For matters of convenience, we first assume that $ATPC$ is drawn uniformly at random from $PTPCS$. In Sect. 5.4, we sketch a modification of our protocol allowing to select an $ATPC$ maximizing the number of traded commodities (without reducing the level of privacy). Other optimization criteria can be integrated analogously.

Definition 6 (Actual Trade). *An actual trade* AT *for an* $ATPC$ *specifies the actual commodities and actual quantities for the commodities traded between the parties involved in* $ATPC$.

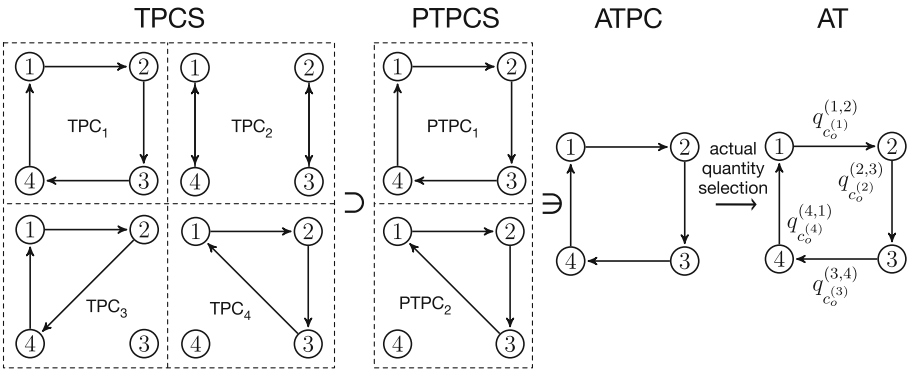

Fig. 1. Illustration of the bartering related terms and their relations.

Figure 1 illustrates the interdependency of the introduced terms. A trade partner constellation can be visualized as a directed graph, i.e., a node represents a party and a directed edge between two nodes represents the exchange direction of a commodity between two parties. For example, according to the node labels and the direction of the edges we have that TPC_4 in Fig. 1 is equal to $(TPT^{(1)}, TPT^{(2)}, TPT^{(3)}, TPT^{(4)}) = ((3,2),(1,3),(2,1),(0,0))$. A potential trade partner constellation set is a subset of a given trade partner constellation set containing those trade partner constellations which form the basis for a trade all involved parties are satisfied with when taking the given context of quotes into account. In Fig. 1, we assume a context of quotes such that $TPC_1 = PTPC_1$ and $TPC_4 = PTPC_2$ are potential trade partner constellations. An actual trade partner constellation is an element from the set of potential trade partner constellation selected w.r.t. a specific strategy. In Fig. 1, the actual trade partner constellation is chosen such that it maximizes the number of traded commodities. The determined actual trade partner constellation is transferred into an actual trade by selecting the actual quantities of the commodities to be traded. In Fig. 1, the actual trade indicates that P_1 has to send $q^{(1,2)}_{c_o^{(1)}}$ units of commodity $c_o^{(1)}$ to P_2, that P_2 has to send $q^{(2,3)}_{c_o^{(2)}}$ units of commodity $c_o^{(2)}$ to P_3, and so on.

4.2 Bartering Process and Intuition

The overall goal of a bartering process between parties $P_1, ..., P_\iota$ with a context of quotes $\mathbf{Q} = (\mathbf{q}^{(1)}, ..., \mathbf{q}^{(\iota)})$ is to determine an actual trade, i.e., one specific trade with which all parties are satisfied. Our bartering process introduced in this paper can determine such an actual trade from the set of all possible trade partner constellations. However, for matters of efficiency, it is also possible to use a smaller trade partner constellation set, e.g., one which may contain only trade partner constellations of 5-trade cycles or constellations in which specific parties get to trade (cf. $TPCS$ in Fig. 1). Upon input of the trade partner constellation

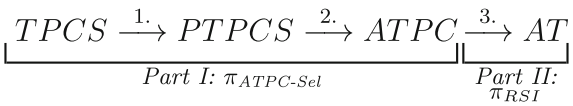

Fig. 2. Illustration of the overall bartering process.

set, the bartering process tries to find an actual trade consistent with the trade partner constellations in the given trade partner constellation set.

Finding an actual trade first requires the determining of the set of potential trade constellations, i.e., those trade constellations in the trade partner constellation set for which the commodities and quantities of the involved parties in their roles of offerer and demander match (Transition 1, Fig. 2). Subsequently, one of the potential trade constellations is selected as actual trade partner constellation (Transition 2, Fig. 2). This constellation then already indicates which parties will send (resp., receive) some commodity to (from) which other party in the (yet to be determined) actual trade. Finally, the parties have the option to individually engage in a two-party protocol with each one of their trade partners (determined by the actual trade partner constellation) in order to select the actual quantities for the commodities to be traded (Transition 3, Fig. 2).

In order to implement such a bartering process securely, the input of the parties, i.e., their quotes, have to be kept secret throughout the process. Moreover, at the end of the process the parties should learn no more than their local view of the selected actual trade, i.e., their own trade partners and the commodities and quantities to be traded with them. Our newly developed bartering process consists of two parts (cf. Fig. 2).

(Part I.) For the first part, we design a secure multi-party protocol $\pi_{\text{ATPC-Sel}}$ that takes a context of private quotes $\mathbf{Q}$ as well as a (publicly known) set of trade partner constellations as input and then performs the following steps: (1) securely determine the potential trade partner constellation set, (2) securely select an actual trade partner constellation, and (3) provide each party P_i with (nothing but) its actual trade partner tuple in the actual trade partner constellation as output (see Sect. 5.2).

(Part II.) In the second part, we propose the option that each party is involved in the two-party protocol π_{RSI} for the secure computation of a random sub-interval (see Sect. 5.3) with each of its trade partners to automatically and fairly determine the actual quantities traded.

For a more comprehensive intuition, we refer to the extended version of this paper [15]. Moreover, the extended version provides an intuition of the novel privacy-preserving mapping operation based on the uniqueness of prime factorization which is used for $\pi_{\text{ATPC-Sel}}$ in order to restrict the output of a party to its local view.

5 Bartering Process

In the following, we introduce our novel multi-party protocol, $\pi_{\text{ATPC-Sel}}$, for selecting an actual trade partner constellation from a given public trade partner constellation set and providing each party with its local view of this actual trade partner constellation as output. We define the underlying functionality $\mathcal{F}_{\text{ATPC-Sel}}$ followed by a detailed protocol description (Sect. 5.2) using the building blocks reviewed in Sect. 5.1. Additionally, in Sect. 5.3 we describe how each party can locally compute their part of the actual trade (i.e., determine the actual quantities for the commodities to be traded) based on the actual trade partner constellation, and how the ATPC-selection can be optimized (Sect. 5.4). In the extended version of this paper [15], we provide an example of how $\pi_{\text{ATPC-Sel}}$ can be used for computing an actual trade from a given trade partner constellation set.

5.1 Building Blocks

Definition 7 ($\mathcal{F}_{\text{OE-TPCF}}$: Oblivious (O) Evaluation (E) of a Trade Partner Constellation Formula (TPCF)). *Let P_i hold private input $\mathbf{q}^{(i)}$ $(i \in \mathscr{P})$ as well as a public trade partner constellation formula $\varphi \in \Phi$. Then, functionality $\mathcal{F}_{OE\text{-}TPCF}$ is given by $(E(e)) \leftarrow \mathcal{F}_{OE\text{-}TPCF}(\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(\iota)}, \varphi)$ where $E(e)$ is an (ι, ι) threshold Paillier ciphertext of $e = 1$ if $[\![\varphi]\!] = 1$ and $e = 0$ otherwise.*

Definition 8 ($\mathcal{F}^{i^*}_{\text{CRS-C}}$: Multi-party Conditional (C) Random (R) Selection (S) with output Check (C)). *Let $P_1, \ldots, P_\iota$ hold m vectors $E(L_i) = (E(l_{i,1}), \ldots, E(l_{i,n}))$ of length n of integers $l_{i,j} \in \mathbb{P}$ $(i \in \mathbb{N}_m,\ j \in \mathbb{N}_n)$ encrypted with (ι, ι) threshold Paillier. Let $E(L_{i^*})$ be an encrypted binary indicator vector and $\{E(L_1), ..., E(L_m)\} \backslash \{E(L_{i^*})\}$ be the value vectors with $i^* \in \mathbb{N}_m$. Then, functionality $\mathcal{F}^{i^*}_{CRS\text{-}C}$ is given by $((E(o_1), \ldots, E(o_m))) \leftarrow \mathcal{F}^{i^*}_{CRS\text{-}C}((E(L_1), \ldots, E(L_m)))$ with $E(o_i) = Rnd(E(l_{i,j^*}))$ $(i \in \mathbb{N}_m)$ where $j^* \leftarrow_{\$} \{j \in \mathbb{N}_n : l_{i^*,j} = 1\}$ if there exists at least one $j \in \mathbb{N}_n$ s.t. $l_{i^*,j} > 0$. Otherwise, $\mathcal{F}^{i^*}_{CRS\text{-}C}((E(L_1), \ldots, E(L_m)))$ outputs $(\lambda_1, \ldots, \lambda_m)$ with $\lambda_1 = \ldots = \lambda_m = \lambda$. Note that j^* is fix for all $i \in \mathbb{N}_m$.*

Definition 9 ($\mathcal{F}^{\omega}_{RSI}$: Two-party secure computation of a Random (R) Sub-Interval (SI)). *Let P_1 hold integer interval I_1 and P_2 hold integer interval I_2 such that $\omega \leq |I_1 \cap I_2|$. Then, functionality $\mathcal{F}^{\omega}_{RSI}$ is given by $([l_r, u_r]) \leftarrow \mathcal{F}^{\omega}_{RSI}(I_1, I_2)$ where $[l_r, u_r]$ is a sub-interval drawn uniformly at random from $I_o = I_1 \cap I_2$ s.t. $|[l_r, u_r]| = \omega$.*

In the extended version of this paper, a novel protocol $\pi_{\text{OE-TPCF}}$ implementing $\mathcal{F}_{\text{OE-TPCF}}$ is introduced (see Sect. 5 in [15]). A protocol implementing functionality $\mathcal{F}^{i^*}_{\text{CRS-C}}$ is presented in [14] and further improved in [17]. A two-party protocol implementing functionality $\mathcal{F}^{\omega}_{RSI}$ is introduced in [5]. All of these protocols have been proven secure in the semi-honest model. In this paper, we exclusively use $\pi^{i^*}_{\text{CRS-C}}$ for $i^* = 1$ and π^{ω}_{RSI} for $\omega = 0$. Consequently, we will omit these indices in the remainder of this paper.

5.2 Protocol for Selecting an Actual Trade Partner Constellation

Definition 10 ($\mathcal{F}_{\text{ATPC-Sel}}$: Actual Trade Partner Constellation (ATPC) Selection (Sel)). *Let party P_i hold private input $\mathbf{q}^{(i)}$ ($i \in \mathscr{P}$). Furthermore, let $TPCS$ be an arbitrary non-empty set of trade partner constellations which is publicly known. Then, the functionality $\mathcal{F}_{ATPC\text{-}Sel}$ is defined as*

$$\left.\begin{array}{ll}(TPT_*^{(1)}, \ldots, TPT_*^{(\iota)}) & \text{if } PTPCS \neq \emptyset \\ \perp & \text{otherwise}\end{array}\right\} \leftarrow \mathcal{F}_{ATPC\text{-}Sel}(\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(\iota)}, TPCS)$$

where $(TPT_*^{(1)}, \ldots, TPT_*^{(\iota)}) := ATPC \leftarrow_{\$} PTPCS \subseteq TPCS$.

In the following, $TPTS^{(i)}$ refers to the set of trade partner tuples for P_i ($i \in \mathscr{P}$) w.r.t. $TPCS$.

In an ideal world where a trusted third party exists, functionality $\mathcal{F}_{\text{ATPC-Sel}}$ could be computed as follows: Each party P_i ($i \in \mathscr{P}$) sends its private input $\mathbf{q}^{(i)}$ to the trusted third party which additionally is given the public set of trade constellation tuples $TPCS$. With the knowledge of $\mathbf{Q} = \{\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(\iota)}\}$, the trusted third party locally computes $PTPCS \subseteq TPCS$. For the case that $PTPCS \neq \emptyset$, the trusted third party selects an actual trade partner constellation $ATPC = (TPT_*^{(1)}, \ldots, TPT_*^{(\iota)})$ uniformly at random from $PTPCS$ and sends $TPT_*^{(i)} = (x_*^{(i)}, y_*^{(i)})$ to P_i. Otherwise, the trusted third party returns $\perp$ to all parties. Note that a $(0, 0)$ output for party P_i indicates that P_i is not involved in the actual trade partner constellation while $\perp$ indicates that there exists no potential trade constellation in the given $TPCS$ at all.

In the real world, where no trusted party exists, protocol $\pi_{\text{ATPC-Sel}}$ (see Protocol 1) is executed in order to compute functionality $\mathcal{F}_{\text{ATPC-Sel}}$. Following the intuition provided in Sect. 4, $\pi_{\text{ATPC-Sel}}$ can be split up into the following phases:

1. *Construction Phase*: From the public set of trade partner constellations, $TPCS$, each party individually constructs the set of formulas Φ such that at the end of this phase each party holds the same set Φ.
2. *Evaluation Phase*: Each $\varphi_j \in \Phi$ is obliviously evaluated jointly by all parties P_i ($i \in \mathscr{P}$) by calling $\pi_{\text{OE-TPCF}}(\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(\iota)}, \varphi_j)$ such that at the end of this phase, each party holds a vector $E(L) = (E(e_1), \ldots, E(e_{|TPCS|}))$ where $e_j = [\![\varphi_j]\!]$ ($j \in \mathbb{N}_{|TPCS|}$).
3. *Mapping Phase*: At the begin of the protocol, each party P_i ($i \in \mathscr{P}$) is given an interval $I^{(i)}$ of positive integers with at least $|TPTS^{(i)}|$ prime numbers such that for each $i, i' \in \mathscr{P}$ ($i \neq i'$), $I^{(i)}$ and $I^{(i')}$ are pairwise disjoint. Each party P_i constructs a secret table mapping each element in $TPTS^{(i)}$ to a unique prime number randomly chosen from $I^{(i)}$. More precisely, each party P_i keeps a set $S^{(i)}$ of already assigned prime numbers from $I^{(i)}$ which is initialized with $\emptyset$. P_i then maps each trade partner tuple $(x^{(i)}, y^{(i)}) \in TPTS^{(i)}$ to a prime number $p^{(i)}_{(x^{(i)}, y^{(i)})} \leftarrow_{\$} \mathbf{P}_{I^{(i)}} \setminus S^{(i)}$. Subsequently, $p_{(x^{(i)}, y^{(i)})}$ is added to $S^{(i)}$. Once all parties have established their mapping tables, all

Protocol 1. $\pi_{\text{ATPC-Sel}}$ for obliviously selecting an actual trade partner constellation.

1 Construction Phase
  1.1 Each party P_i ($i \in \mathscr{P}$) locally constructs the same set Φ from $TPCS$.
2 Evaluation Phase
  2.1 For each $\varphi_j \in \Phi$:
    2.1.1 Each party P_i participates in $(E(e_j)) \leftarrow \pi_{\text{OE-TPCF}}(\varphi_j)$
  2.2 Each party P_i sets $E(L) := (E(e_1), \ldots, E(e_{|TPCS|}))$
3 Mapping Phase
  3.1 Each party P_i:
    3.1.1 Set $S^{(i)} := \emptyset$
    3.1.2 For each $(x^{(i)}, y^{(i)}) \in TPTS^{(i)}$:
      3.1.2.1 Draw a random prime $p^{(i)}_{(x^{(i)},y^{(i)})}$ from $\mathbf{P}_{I^{(i)}} \backslash S^{(i)}$
      3.1.2.2 Update $S^{(i)} = S^{(i)} \cup \{p^{(i)}_{(x^{(i)},y^{(i)})}\}$
  3.2 Party P_ι:
    3.2.1 Set $u^{(\iota)}_j := E(p^{(\iota)}_{TPT^{(\iota)}_j})$ $(\varphi_j \overset{\mathfrak{Q}}{\approx} TPC_j)$
    3.2.2 Send $(u^{(\iota)}_1, \ldots, u^{(\iota)}_{|TPCS|})$ to $P_{\iota-1}$
  3.3 Each party $P_{i'}$ (from $i' = \iota - 1$ to 1)
    3.3.1 Compute $u^{(i')}_j := u^{(i'+1)}_j \times_h p^{(i')}_{TPT^{(i')}_j} +_h E(0)$
    3.3.2 Send $(u^{(i')}_1, \ldots, u^{(i')}_{|TPCS|})$ to $P_{i'-1}$
  3.4 Party P_1:
    3.4.1 Set $E(L') := (E(e'_1), \ldots, E(e'_{|TPCS|})) := (u^{(1)}_1, \ldots, u^{(1)}_{|TPCS|})$
    3.4.2 Broadcast $E(L')$
4 Selection Phase
  4.1 Each party P_i participates in $((c^*_1, c^*_2)) \leftarrow \pi_{\text{CRS-C}}(E(L), E(L'))$
  4.2 For each party P_i
    4.2.1 If $c^*_1 = c^*_2 = \lambda$:
      4.2.1.1 Skip Steps 5 to 7
      4.2.1.2 Each party P_i outputs $\perp$
5 Decryption Phase
  5.1 All parties jointly compute $e^*_2 = D(c^*_2)$
6 Reverse Mapping Phase
  6.1 Each Party P_i:
    6.1.1 For each $p^{(i)}_{TPT^{(i)}_j} \in S^{(i)}$
      6.1.1.1 If $p^{(i)}_{TPT^{(i)}_j}$ divides e^*_2 then $TPT^{(i)}_* := TPT^{(i)}_j$ and go to Step 7
7 Output Phase
  7.1 Each party P_i outputs $TPT^{(i)}_*$

parties engage in the consecutive computation of an encrypted prime number product for each $\varphi_j \in \Phi$. Each party P_i contributes a single prime number $p^{(i)}_{TPT^{(i)}_j}$ to the encrypted prime number product associated with $\varphi_j \in \Phi$:

First, P_ι computes $u_j^{(\iota)} = E(p^{(\iota)}_{TPT_j^{(\iota)}})$ ($j \in \mathbb{N}_{|TPCS|}$) and sends the result to $P_{\iota-1}$. Each party $P_{i'}$ from $i' = \iota - 1$ to 1 then computes $u_j^{(i')} = u_j^{(i'+1)} \times_h p_{TPT_j^{(i')}} +_h E(0)$ and sends the results to $P_{i'-1}$, except P_1 which sets $E(L') := (E(e'_1), \ldots, E(e'_{|TPCS|})) := (u_1^{(1)}, \ldots, u_{|TPCS|}^{(1)})$ and broadcasts $E(L')$. This mapping of trade partner constellations to prime number products is one of the central ideas of this protocol and ensures the security of the protocol.

4. *Selection Phase*: From the previous phases, each φ_j is associated with two values $E(e_j)$ and $E(e'_j)$ where $e_j \in \{0, 1\}$ indicates whether or not φ_j is satisfied while e'_j is a product of individual prime numbers encoding the trade partner tuples of each party w.r.t. φ_j. In this phase, the parties now jointly compute $\pi_{\text{CRS-C}}$ on the common input $(E(L), E(L'))$ in order to select an entry of $E(L')$ associated with a randomly selected $\varphi_j \in \Phi_{sat}$ for the case that $\Phi_{sat} \neq \emptyset$ (i.e., $PTPCS \neq \emptyset$). Otherwise, in the case that $\Phi_{sat} = \emptyset$ (i.e., $PTPCS = \emptyset$), the parties learn of this fact. In the former case, $\pi_{\text{CRS-C}}$ returns a randomly selected pair $(c_1^*, c_2^*) \in (E(L), E(L'))$ with $c_1^* = E(e_1^*)$ and $c_2^* = E(e_2^*)$. In the latter case where $e_1 = \ldots = e_{|TPCS|} = 0$, $\pi_{\text{CRS-C}}(L, L')$ returns (c_1^*, c_2^*) with $c_1^* = c_2^* = \lambda$ which prompts each party P_i to output $\perp$ and to terminate the protocol. The purpose for this approach is to hide the number of satisfied formulas (for the case that $\Phi_{sat} \neq \emptyset$) as this could otherwise not be simulated given the inputs and outputs of the set of corrupted parties.
5. *Decryption Phase*: Each party learns e_2^* from jointly decrypting c_2^* together with all other parties.
6. *Reverse Mapping Phase*: Each party P_i checks which prime in $S^{(i)}$ divides e_2^*. The unique result $TPT_*^{(i)}$ determines P_i's trade partners w.r.t. $\varphi \overset{\mathfrak{Q}}{\sim} ATPC$.
7. *Output Phase*: Each party P_i outputs $TPT_*^{(i)}$.

Theorem 1. *Let P_i hold $\mathbf{q}^{(i)}$ ($i \in \mathscr{P}$) and let $TPCS$ be public. Then protocol $\pi_{ATPC\text{-}Sel}$ securely computes functionality $\mathcal{F}_{ATPC\text{-}Sel}$ in the semi-honest model.*

Proof (COD). In order to prove COD, we distinguish two cases: (i) $TPCS \supseteq PTPCS = \emptyset$ and (ii) $TPCS \supseteq PTPCS \neq \emptyset$. For case (i), the output of $\pi_{\text{ATPC-Sel}}$ is fixed; each party outputs $\perp$. For case (ii), we have to show $ATPC = (TPT_*^{(1)}, \ldots, TPT_*^{(\iota)})$ is selected uniformly at random from $PTPCS$.

(i) For the case that $TPCS \supseteq PTPCS = \emptyset$, the Evaluation Phase of $\pi_{\text{ATPC-Sel}}$ returns a vector $E(L) = (E(e_1), \ldots, E(e_{|TPCS|}))$ where $e_1 = \ldots = e_{|TPCS|} = 0$ since there exists no $\varphi \in \Phi$ such that $[\![\varphi]\!] = 1$. This implies that in the Selection Phase of $\pi_{\text{ATPC-Sel}}$, $\pi_{\text{CRS-C}}(E(L), E(L'))$ returns (λ, λ). Then, each party P_i ($i \in \mathscr{P}$) outputs $\perp$ and the protocol terminates.

(ii) For the case that $TPCS \supseteq PTPCS \neq \emptyset$, the Evaluation Phase of $\pi_{\text{ATPC-Sel}}$ computes a vector $E(L) = (E(e_1), \ldots, E(e_{|TPCS|}))$ with $e_j = [\![\varphi_j]\!]$ and L has Hamming weight $|\Phi_{sat}|$. $\pi_{\text{CRS-C}}(E(L), E(L'))$, called in the Selection Phase of $\pi_{\text{ATPC-Sel}}$, returns $(E(e_1^*), E(e_2^*))$ for a random $j \in \mathbb{N}_{|TPCS|}$ such that $e_1^* = e'_j = [\![\varphi_j]\!] = 1$ ($\varphi_j \in \Phi_{sat}$) and $e_2^* = p^{(1)}_{TPT_j^{(1)}} \cdot \ldots \cdot p^{(\iota)}_{TPT_j^{(\iota)}}$. After

jointly decrypting $E(e_2^*)$ in the Decryption Phase of $\pi_{\text{ATPC-Sel}}$, each party P_i obtains e_2^* and sets its $TPT_*^{(i)} := TPT_j^{(i)}$ for $p_{TPT_j^{(i)}}^{(i)} \in S^{(i)}$ where $p_{TPT_j^{(i)}}^{(i)}$ divides e_2^*. Overall, it follows that $ATPC \leftarrow_{\$} PTPCS \subseteq TPCS$.

(CVD). By separating the different phases of Protocol 1, we sketch a simulator $\mathcal{S}$ which outputs a transcript computationally indistinguishable from $\text{VIEW}_{I_C}^{\pi}(\widehat{X})$. A detailed description of $\mathcal{S}$ is provided by [15]. Note that the number of messages a party P_i ($i \in \mathscr{P}$) receives when participating in $\pi_{\text{ATPC-Sel}}$ depends on the party's position in the protocol execution and on whether or not $PTPCS = \emptyset$. By applying the modular composition theorem, it suffices for $\mathcal{S}$ to simulate the output of $\pi_{\text{OE-TPCF}}$ and $\pi_{\text{CRS-C}}$ by means of a trusted third party performing the computation of $\mathcal{F}_{\text{OE-TPCF}}$ and $\mathcal{F}_{\text{CRS-C}}$, respectively. In the Evaluation Phase of $\pi_{\text{ATPC-Sel}}$, the joint outputs of the $|TPCS|$ sub-protocol calls of $\pi_{\text{OE-TPCF}}$ are simulated by setting $\langle E(L)\rangle := (\langle E(e_1)\rangle, \ldots, \langle E(e_{|TPCS|})\rangle)$ where $\langle E(e_j)\rangle \leftarrow_{\$} \mathbb{C}$ ($j \in \mathbb{N}_{|TPCS|}$). For each P_c ($c \in I_C = \{i_1, \ldots, i_\kappa\}$), $\mathcal{S}$ simulates $E(L)$ as $\langle E(L)\rangle$. The Mapping Phase can be simulated by performing Steps 3.1–3.3 of Protocol 1 for each party P_i ($i \in \mathscr{P}$). From the simulated mapping tables, $\mathcal{S}$ simulates $(u_1^{(i)}, \ldots, u_{|TPCS|}^{(i)})$ for $P_{c'}$ with $c' \in I_C \backslash \{\iota\}$ (cf. Steps 3.2 and 3.3, Protocol 1) and $E(L')$ for $P_{c''}$ with $c'' \in I_C \backslash \{1\}$ (cf. Step 3.4, Protocol 1). Furthermore, $\mathcal{S}$ computes $\langle e_j'\rangle$ ($j \in \mathbb{N}_{|TPCS|}$) from the simulated mapping tables (where one of these values is used to simulate the Decryption Phase). The simulation of the Selection Phase depends on whether or not $\mathcal{F}(\widehat{X}) = \perp$. For the case that $\mathcal{F}(\widehat{X}) \neq \perp$, the output of $\pi_{\text{CRS-C}}$ is simulated by setting $\langle c_1^*\rangle, \langle c_2^*\rangle \leftarrow_{\$} \mathbb{C}$. Otherwise, $\langle c_1^*\rangle = \langle c_2^*\rangle := \lambda$. The Decryption Phase is only executed for the case that $\mathcal{F}(\widehat{X}) \neq \perp$. The output of the decryption protocol D is simulated by setting $\langle e_2^*\rangle := \langle e_j'\rangle$ where $j \in \mathbb{N}_{|TPCS|}$ is chosen such that $\varphi_j \overset{\mathfrak{Q}}{\sim} TPC_j$ with $TPT_j^{(i_1)} = TPT_*^{(i_1)}, \ldots, TPT_j^{(i_\kappa)} = TPT_*^{(i_\kappa)}$. Note that otherwise, $\langle e_2^*\rangle$ is not consistent with $\mathcal{F}(\widehat{X})$. Due to the fact that the underlying cryptosystem is semantically secure, it follows that the simulated view is computationally indistinguishable from $\text{VIEW}_{I_C}^{\pi}$.

Complexity. Let $O_{\text{OE-TPCF}}$, $O_{\text{CRC-C}}$, and O_{Dec} denote the computation, communication, and round complexities of $\pi_{\text{OE-TPCF}}$, $\pi_{\text{CRS-C}}$, $D(\cdot)$, respectively, depending on the context. The computation complexity of $\pi_{\text{ATPC-Sel}}$ is dominated by the sub-protocol calls and $|TPCS|$ homomorphic scalar multiplications and overall is in $\mathcal{O}(|TPCS| + |TPCS| \cdot O_{\text{OE-TPCF}} + O_{\text{CRC-C}} + O_{\text{Dec}})$. The communication complexity of $\pi_{\text{ATPC-Sel}}$ is in $\mathcal{O}(\iota \cdot |TPCS| + |TPCS| \cdot O_{\text{OE-TPCF}} + O_{\text{CRC-C}} + O_{\text{Dec}})$ while the round complexity is in $\mathcal{O}(\iota + |TPCS| \cdot O_{\text{OE-TPCF}} + O_{\text{CRC-C}} + O_{\text{Dec}})$.

5.3 Negotiation of Actual Quantities

In order to complete the (privacy-preserving) bartering process, i.e., for each party to compute its local view of the AT based on the $ATPC$, each party has

to negotiate the actual quantities of the commodities to be traded with its trade partner. This can be done either offline without any privacy-preserving protocol or, e.g., by engaging in the two-party protocol π_{RSI} with each one of its trade partners. That is for each $TPT_*^{(i)} = (x_*^{(i)}, y_*^{(i)}) \neq (0,0)$, P_i and $P_{y^{(i)}}$ participate in an execution of $(q_{c_o^{(i)}}^{(i,y^{(i)})}) \leftarrow \pi_{RSI}(Q_d^{(y^{(i)})}, Q_o^{(i)})$ where $q_{c_o^{(i)}}^{(i,y^{(i)})}$ indicates the quantity of $c_o^{(i)}$ that P_i has to send to $P_{y^{(i)}}$. Note that $q_{c_o^{(i)}}^{(i,y^{(i)})}$ is chosen uniformly at random from $Q_d^{(y^{(i)})} \cap Q_o^{(i)}$ which honors the specified quantity ranges of the parties without preferring any one of them. Alternatively, to avoid possible imbalances in regards to quantity selection (for details see [16]), it is possible to shrink the private overlap interval $Q_d^{(y^{(i)})} \cap Q_o^{(i)}$ obliviously around the midpoint of the interval (using a similar approach to the one described in [16]).

5.4 Optimization of ATPC-Selection

Until now, we assumed that $ATPC$ was drawn uniformly at random from the set of potential trade partner constellations $PTPCS$. We now sketch a simple modification of protocol $\pi_{\text{ATPC-Sel}}$ which allows the private selection of an $ATPC$ with maximum *welfare* as optimization criteria, where the welfare $\mathcal{W}(\cdot)$ of a TPC is defined as the number of parties actively involved in the trade: $\mathcal{W}(TPC) := |\{TPT^{(i)} : i \in \mathscr{P}, TPT^{(i)} \in TPC, TPT^{(i)} \neq (0,0)\}|$.

The first step of our protocol modification is to introduce a prioritization of the TPCs given by $TPCS$: At the end of the Evaluation Phase (Step 2.2 in Protocol 1), the parties locally multiply the evaluation result $E(e_i)$ with $\mathcal{W}(TPC_i)$ $(\forall i \in \mathbb{N}_{|TPCS|})$ resulting in a vector $E(L) = (E(e_1) \times_h \mathcal{W}(TPC_1),$ $\ldots, E(e_{|TPCS|}) \times_h \mathcal{W}(TPC_{|TPCS|}))$. The second step of our modification is to replace the protocol call of $\pi_{\text{CRS-C}}$ (Step 4.1, Protocol 1) by a variant of conditional random selection (also introduced in [14]) which supports an integer indicator vector instead of just a binary indicator vector (cf. Definition 8). In the context of Protocol 1, this variant of $\pi_{\text{CRS-C}}$ returns $(c_1^*, c_2^*) := (E(e_{j^*}), E(e'_{j^*}))$ where $j^* \leftarrow_\$ \{j \in \mathbb{N}_{|TPCS|} : e_j = max(e_1, \ldots, e_{|TPCS|})\}$.

Similar optimization criteria (e.g., for each TPC given by $TPCS$ a party individually determines the corresponding utility value and the welfare of a given TPC corresponds to the sum of utility values over all parties) can be integrated into protocol $\pi_{\text{ATPC-Sel}}$ analogously.

Acknowledgments. This work was supported by DFG Award ME 3704/4-1.

References

1. Aïmeur, E., Brassard, G., Mani Onana, F.S.: Blind sales in electronic commerce. In: Proceedings of the 6th International Conference on Electronic Commerce, pp. 148–157. ACM (2004)
2. Canetti, R.: Security and composition of multiparty cryptographic protocols. J. Cryptol. **13**(1), 143–202 (2000)

3. Damgård, I., Jurik, M.: A Generalisation, a Simplification and some applications of paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_9
4. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
5. Förg, F., Mayer, D., Wetzel, S., Wüller, S., Meyer, U.: A secure two-party bartering protocol using privacy-preserving interval operations. In: 12th Annual International Conference on Privacy, Security and Trust, pp. 57–66 (2014)
6. Franklin, M., Tsudik, G.: Secure group barter: multi-party fair exchange with semi-trusted neutral parties. In: Hirchfeld, R. (ed.) FC 1998. LNCS, vol. 1465, pp. 90–102. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055475
7. Frikken, K., Opyrchal, L.: PBS: private bartering systems. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 113–127. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85230-8_9
8. Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge (2009)
9. Kannan, S., Morgenstern, J., Rogers, R., Roth, A.: Private pareto optimal exchange. In: Proceedings of the Sixteenth ACM Conference on Economics and Computation, pp. 261–278. ACM (2015)
10. López, N., Núñez, M., Rodríguez, I., Rubio, F.: A multi-agent system for e-barter including transaction and shipping costs. In: Proceedings of the 2003 ACM Symposium on Applied Computing, pp. 587–594. ACM (2003)
11. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
12. Nzouonta, J., Silaghi, M.-C., Yokoo, M.: Secure computation for combinatorial auctions and market exchanges. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1398–1399. IEEE Computer Society (2004)
13. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
14. Wüller, S., Meyer, U., Förg, F., Wetzel, S.: Privacy-preserving conditional random selection (extended version). In: 13th Annual Conference on Privacy, Security and Trust, pp. 44–53 (2015)
15. Wüller, S., Meyer, U., Wetzel, S.: Towards privacy-preserving multi-party bartering (extended version). Technical report AIB-2016-10, RWTH Aachen (2016)
16. Wüller, S., Pessin, W., Meyer, U., Wetzel, S.: Privacy-preserving two-party bartering secure against active adversaries. In: 14th Annual Conference on Privacy, Security and Trust, pp. 229–238 (2016)
17. Wüller, S., Mayer, D., Förg, F., Schüppen, S., Assadsolimani, B., Meyer, U., Wetzel, S.: Designing privacy-preserving interval operations based on homomorphic encryption and secret sharing techniques. J. Comput. Secur. **25**(1), 59–81 (2017)

Springer

http://www.springer.com/978-3-319-70277-3

Financial Cryptography and Data Security
FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers
Brenner, M.; Rohloff, K.; Bonneau, J.; Miller, A.; Ryan, P.Y.A.; Teague, V.; Bracciali, A.; Sala, M.; Pintore, F.; Jakobsson, M. (Eds.)

2017, XXII, 636 p. 97 illus., Softcover
ISBN: 978-3-319-70277-3