Efficient Privacy-Preserving Protocols for Multi-Unit Auctions

Felix Brandt and Tuomas Sandholm

[1] Stanford University, Stanford CA 94305, USA
brandtf@cs.stanford.edu
[2] Carnegie Mellon University, Pittsburgh PA 15213, USA
sandholm@cs.cmu.edu

Abstract. The purpose of multi-unit auctions is to allocate identical units of a single type of good to multiple agents. Besides well-known applications like the selling of treasury bills, electrical power, or spectrum licenses, multi-unit auctions are also well-suited for allocating CPU time slots or network bandwidth in computational multiagent systems. A crucial problem in sealed-bid auctions is the lack of trust bidders might have in the auctioneer. For one, bidders might doubt the correctness of the auction outcome. Secondly, they are reluctant to reveal their private valuations to the auctioneer since these valuations are often based on sensitive information. We propose privacy-preserving protocols that allow bidders to jointly compute the auction outcome without the help of third parties. All three common types of multi-unit auctions (uniform-price, discriminatory, and generalized Vickrey auctions) are considered for the case of marginal decreasing valuation functions. Our protocols are based on distributed homomorphic encryption and can be executed in a constant number of rounds in the random oracle model. Security merely relies on computational intractability (the decisional Diffie-Hellman assumption). In particular, no subset of (computationally bounded) colluding participants is capable of uncovering private information.

1 Introduction

Auctions are not only wide-spread mechanisms for selling goods, they have also been applied to a variety of computer science settings like task assignment, bandwidth allocation, or finding the shortest path in a network with selfish nodes. A crucial problem in sealed-bid auctions is the lack of trust bidders might have in the auctioneer. For one, bidders might doubt the correctness of the auction outcome. Secondly, they are reluctant to reveal their private valuations to the auctioneer since these valuations are often based on sensitive information. We tackle both problems by providing cryptographic protocols that allow bidders to jointly compute the auction outcome without revealing any other information.

More specifically, our setting consists of one seller and n bidders that intend to come to an agreement on the selling of M indistinguishable units of a particular

type of good.[3] Each bidder submits a vector of M sealed bids $(b_1^i, b_2^i, \ldots, b_M^i)$ to the auctioneer, expressing how much he is willing to pay for each additional unit. In other words, $\sum_{j=1}^{m} b_j^i$ is the amount bidder i is willing to pay for m units. A common assumption that we also make is that bidders have *marginal decreasing valuations*, *i.e.*, $b_1^i \geq b_2^i \geq \cdots \geq b_M^i$. This is justified by the fact that bidders usually want to pay less for each additional unit the more units they already have.[4] The auctioneer then clears the auction by allocating units to the bidders that value them most. Let W be the set of winning bids, *i.e.*, the set containing the M highest bids. Clearly, an economically efficient auction should allocate m^i items to bidder i if he submitted m^i bids that belong to W. There are three common ways of pricing units that are sold in multi-unit auctions: uniform-price, discriminatory, and generalized Vickrey (see *e.g.*, [Kri02] or [Kle99]).

- *Uniform-Price Auction*
  All bidders pay the same price per unit, given by the $(M+1)$st-highest bid.
- *Discriminatory Auction*
  The discriminatory auction is the natural extension of the 1st-price sealed-bid auction (for one unit) to the case of M units. Every bidder pays exactly what he bid for each particular unit he receives. In other words, if bidder i receives m^i units, he pays $\sum_{m=1}^{m^i} b_m^i$.
- *Generalized Vickrey Auction*
  The generalized Vickrey auction is an extension of the Vickrey (or 2nd-price sealed-bid) auction. A bidder that receives m units pays the sum of the m highest losing bids submitted by other bidders, *i.e.*, excluding his own losing bids. This auction format belongs to the praised family of VCG mechanisms [Vic61,Cla71,Gro73] and provides various desirable theoretical properties.

There is an ongoing debate in economic theory which auction format is most favorable. For example, the uniform-price auction is sometimes rejected because it suffers from an effect called *demand reduction* which states that bidders are better off reducing their bids for additional units. In contrast to both other auction types, the generalized Vickrey auction is *economically efficient*, *i.e.*, the total welfare of all bidders is maximized in a strategic equilibrium, and *strategy-proof*, *i.e.*, each bidder is best off bidding his true valuations no matter what other bidders do. On the other hand, the generalized Vickrey auction is vulnerable to strategic collusion and can result in outcomes that might be considered unfair. Summing up, it seems as if different application scenarios require different auction types. For example, the US government began to use uniform-price auctions to sell treasury bills in 1992, after a long tradition of discriminatory auctions. On the other hand, UK electricity generators switched from uniform-price to discriminatory auctions in 2000. A detailed discussion of the pros and cons of

[3] All the presented protocols also work for procurement or so-called reverse auctions where there is one buyer and multiple sellers.

[4] However, this is not always the case. For instance, in a tire auction, a car owner might value the forth tire higher than the third.

multi-unit auctions is beyond the scope of this paper (see *e.g.*, [Kri02] or [Kle99] for further information).

In this paper, we propose cryptographic protocols for all three common types of multi-unit auctions. These protocols allow bidders to "emulate" a virtual auctioneer, thus enabling privacy of bids without relying on third parties. The only information revealed in addition to the auction outcome is minor statistical data in the case of certain ties (*e.g.*, the number of tied bids). As round efficiency is usually considered to be the most important complexity measure in a distributed setting, the main goal when designing these protocols was to minimize the number of rounds required for executing the protocols. In fact, all our protocols only need a low constant number of rounds in the random oracle model. Communication and computation complexity, on the other hand, is linear in the number of different prices. Nevertheless, the proposed protocols should be practically feasible for moderately sized scenarios.

The remainder of this paper is structured as follows. In Section 2, we describe the general security model underlying this work. Recent related research on cryptographic auction protocols is reviewed in Section 3. In Section 4, we give a detailed description of the vector notation and order statistic subprotocol to be used in the multi-unit auction protocols presented in Section 5. Concrete implementation details regarding El Gamal encryption and efficient (honest-verifier) zero-knowledge proofs are discussed in Section 6. The paper concludes with an overview of the obtained results in Section 7.

2 Security Model

Our primary goal is privacy that cannot be broken by any coalition of third parties or bidders. For this reason, we advocate a security model in which bidders themselves jointly compute the auction outcome so that any subset of bidders is incapable of revealing private information. Clearly, extensive interaction by bidders is undesirable in practice (but unavoidable given our objective). In order to minimize interaction, our secondary goal is to keep round complexity at a minimum (small constants). The main drawbacks implied by our setting are weak robustness and high computational and communication complexity. However, auctions that require such a high degree of privacy typically take place with few, well-known (*i.e.*, non-anonymous) bidders, for instance when auctioning off spectrum licenses.

We consider cryptographic protocols for n bidders and one seller. Each bidder possesses a private input consisting of M bids. Agents engage in a multi-party protocol to jointly and securely compute the outcome function f. In our context, security consists of correctness (f is computed correctly) and full privacy (aka. $(n-1)$-privacy, *i.e.*, no subset of agents learns more information than what can be inferred from the outcome and the colluding agents' private inputs). When allowing premature protocol abort, any such function f can be computed securely and fairly when trapdoor permutations exist, and a designated agent

does not quit or reveal information prematurely.[5] In the auction protocols presented in this paper, the seller will take the role of the designated agent. It is important to note that even when the seller quits or reveals information early, the worst thing that can happen is that an agent learns the outcome and quits the protocol before the remaining agents were able to learn the outcome.[6] Bid privacy is not affected by premature abort.

Whenever a malicious bidder disrupts the protocol by sending faulty messages or failing to prove the correctness of his behavior in zero-knowledge, this bidder will be removed, and the protocol will be restarted (termination is guaranteed after at most $n-1$ iterations). We presume that the "public" is observing the protocol and therefore a malicious bidder can undoubtedly be identified, independently of how many remaining agents are trustworthy. As malicious bidders can easily be fined and they do not gain any information, there should be no incentive to disrupt the auction and we henceforth assume that a single protocol run suffices.

3 Related Work

Numerous cryptographic protocols for *single-unit* auctions have been proposed in the literature (*e.g.*, [AS02,BS01,Bra03a,Di 00,JS02,Kik01,LAN02,NPS99]). We follow our previous approach [Bra03a] where bidders jointly compute the auction outcome without the help of trusted third parties. There are few privacy-preserving protocols for the selling of more than just a single good. Suzuki et al [SY02,SY03,YS04] proposed protocols for general *combinatorial auctions* (see *e.g.*, [CSS05]), where bidders can bid on arbitrary combinations of items for sale, based on a secure dynamic programming subprotocol. The problem of determining the winners in this type of auction is $\mathcal{NP}$-complete. Clearly, adding cryptographic overhead to winner determination results in protocols whose complexity is prohibitively large for most practical settings. Multi-unit auctions, in which a specific number of identical units of a single item is sold, are an important, yet still intractable [SS01], subcase of combinatorial auctions. Instead of bidding on every conceivable combination of items, bidders simply specify their willingness to pay for any number of units. In contrast to general combinatorial auctions, multi-unit auctions are already widely used, *e.g.*, for selling treasury bills, electrical power, or spectrum licenses. Suzuki et al formulate the winner determination problem in multi-unit auctions as a dynamic programming optimization problem, thus enabling their secure dynamic programming protocol to compute the optimal allocation of units [SY02,SY03]. However, when making the reasonable assumption that bidders' valuations are *marginal decreasing* in the number of units, *i.e.*, the $(m+1)$th unit a bidder receives is never more

[5] This useful restriction to circumvent fairness problems was also used in our previous work (*e.g.*, [Bra03b,Bra03a]). Independently, the security of such a model was generally analyzed by Goldwasser et al [GL02].

[6] Another common way to obtain fairness without a trusted majority is the gradual release of secrets (*e.g.*, [Yao86,GL90].

valuable to him than the mth unit, computing the optimal allocation of units becomes tractable [Ten00], thus making computationally demanding techniques like dynamic programming unnecessary. To the best of our knowledge, cryptographic protocols for multi-unit auctions with marginal decreasing valuations have only been presented for the considerably simple subcase where each bidder only demands a single unit [AS02,Bra03a,Kik01].[7]

Parallel to our work on fully private auction and social choice protocols (*e.g.*, [Bra02,Bra03b,BS04b,BS04a]), there is an independent, yet quite similar, stream of research on self-tallying elections [KY02,KY03,Gro04]. In both settings, agents jointly determine the outcome of a social choice function without relying on trusted third parties. What we call "full privacy" is termed "perfect ballot secrecy" in Kiayias et al's work. Similarly, the terms "self-tallying" and "dispute-free" [KY02] can be translated to "bidder-resolved" and "weakly robust" [Bra02], respectively. In order to achieve fairness, both approaches assume a weakly trustworthy party (a "dummy bidder" and the auction seller, respectively). Besides these similarities, Kiayias et al's approach mainly differs in the emphasis of non-interactiveness (once the random-generating preprocessing phase is finished) while computing rather simple outcome functions (*e.g.*, the sum of input values).

4 Building Blocks

Distributed homomorphic encryption allows agents to efficiently add secret values without extensive interaction. For this reason, our protocols only require the computation of *linear combinations* of secret inputs values (which can be solely based on addition) and multiplications with jointly created random numbers (for which we propose an efficient sub-protocol in Section 6.1). When computing on *vectors* of secrets, the computation of linear combinations enables the addition (and substraction) of secret vectors, and the multiplication of vectors with predefined known matrices. Furthermore, the vector representation allows for efficient zero-knowledge proofs of correctness.

4.1 Vector Representation

Let $\boldsymbol{p}$ be a vector of k possible prices (or valuations), $\boldsymbol{p} = (p_1, p_2, \ldots, p_k)$, and $bid \in \{1, 2, \ldots, k\}$ a bid. The *bid vector* $\boldsymbol{b}$ of this bid is defined so that component $b_{bid} = 1$ (the bidder bids p_{bid}) and all other components are 0. This representation allows efficient proofs of the vector's correctness by showing $\forall j \in \{1, 2, \ldots, k\} : \; b_j \in \{0, 1\}$ and $\sum_{j=1}^{k} b_j = 1$ (see Section 6 for details). Yet, the main advantage of the vector representation is the possibility to efficiently perform certain computations. For example, the "integrated" bid vector $\boldsymbol{b'}$ (a

[7] In this so-called *unit demand* case, the uniform-price and the generalized Vickrey auction collapse to the same auction type: the $(M+1)$st-price auction.

notion introduced in [AS02]) can be derived by multiplying the bid vector with the $k \times k$ lower triangular matrix L.[8]

$$\boldsymbol{b} = \begin{pmatrix} b_k \\ \vdots \\ b_{bid-1} \\ b_{bid} \\ b_{bid+1} \\ \vdots \\ b_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \boldsymbol{b}' = \mathsf{L}\,\boldsymbol{b} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \text{ where } \mathsf{L} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix}$$

The price we pay for round-efficiency enabled by this unary representation is communication and computation complexity that is linear in the number of different prices k. On the other hand, the unary notation allows us to easily adapt the given protocols to emulate *iterative* (*e.g.*, ascending-price or descending-price) auctions (see *e.g.*, Chapter 2 of [CSS05]) in which bidders gradually express their unit demand for sequences of prices. In fact, there are common iterative equivalences for each of the three sealed-bid auction mechanisms considered in this paper: the multi-unit English auction (uniform-price), the multi-unit Dutch auction (discriminatory), and the Ausubel auction (generalized Vickrey). Iterative auctions are sometimes preferred over sealed-bid auctions because bidders are not required to exhaustively determine their valuations and because they can lead to higher revenue if valuations are inter-dependent.

4.2 Order Statistic Subprotocol

The most essential building block of our auction protocols is a subprotocol that determines the mth order statistic, *i.e.*, the mth highest bid, in a given vector of N bids. Some $k \times k$ matrices that we will use in addition to L are the upper triangular matrix U, the identity matrix I, and random multiplication matrices R^*. Furthermore, we will utilize the k-dimensional unit vector $\boldsymbol{e}$.

$$\mathsf{U} = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \mathsf{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \quad \mathsf{R}^* = \begin{pmatrix} * & 0 & \cdots & 0 \\ 0 & * & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & * \end{pmatrix}, \quad \boldsymbol{e} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

The components on the diagonal of R^* are random numbers unknown to the agents. They are jointly created using a special sub-protocol. Multiplication with R^* turns all vector components *that are not zero* into meaningless random numbers. For this reason, it is usually a final masking step in our protocols.

[8] Please note that matrices are only used to facilitate the presentation. The special structure of all used matrices allows us to compute matrix-vector multiplications in $\mathcal{O}(k)$ steps.

Our approach to detect the mth-highest bid requires special techniques if there is a tie at the mth-highest bid. Information that is revealed in case of a tie is the number of tied bids (t) and the number of bids that are greater than the mth-highest bid (u). Let us for now assume that there is always a *single* mth-highest bid ($t = 1$ and $u = m - 1$). When given vector $\boldsymbol{B}$ where each component of $\boldsymbol{B}$ denotes the number of bids at a specific price (see Example 1), we will specify how to compute a vector that merely reveals the mth-highest bid.

$$\boldsymbol{stat}^m_{1,m-1}(\boldsymbol{B}) = \Big((2\mathsf{L} - \mathsf{I})\boldsymbol{B} - (2m-1)\boldsymbol{e}\Big)\mathsf{R}^*$$

yields a vector in which the component denoting the mth-highest bid is zero. All other components are random values.

Example 1. Let the vector of possible prices be $\boldsymbol{p} = (10, 20, 30, 40, 50, 60)$ and consider the computation of the second-highest bid ($m = 2$) in a vector that represents bids 20 and 50:

$$\boldsymbol{B} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

All computations take place in the finite field $\mathbb{Z}_{11}$. Asterisks denote arbitrary random numbers that have no meaning to bidders.

$$\boldsymbol{stat}^2_{1,1}(\boldsymbol{B}) = \left(\begin{pmatrix} 1&0&0&0&0&0 \\ 2&1&0&0&0&0 \\ 2&2&1&0&0&0 \\ 2&2&2&1&0&0 \\ 2&2&2&2&1&0 \\ 2&2&2&2&2&1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{pmatrix}\right)\mathsf{R}^* = \left(\begin{pmatrix} 0 \\ 1 \\ 2 \\ 2 \\ 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{pmatrix}\right)\mathsf{R}^* = \begin{pmatrix} 8 \\ 9 \\ 10 \\ 10 \\ 0 \\ 1 \end{pmatrix}\mathsf{R}^* = \begin{pmatrix} * \\ * \\ * \\ * \\ 0 \\ * \end{pmatrix}$$

The resulting vector $\boldsymbol{stat}^2_{1,1}(\boldsymbol{B})$ indicates that the second-highest bid is 20. ▲

When two or more bids qualify as the mth-highest bid (because they are equal), the technique described above does not work ($\boldsymbol{stat}^m_{1,m-1}(\boldsymbol{B})$ contains no zeros). For this reason, we compute additional vectors that yield the correct outcome in the case of such a tie. The following method marks the mth-highest bid while not revealing any information about other ties. Subtracting $t\boldsymbol{e}$ from input vector $\boldsymbol{B}$ yields a vector that contains zeros if there is a tie of t bids ($1 < t \leq N$ where N is the number of bids). As we are only interested in ties at the mth-highest bid, other ties are masked by adding $(N+1)\,(\mathsf{L}\,\boldsymbol{B} - (t+u)\boldsymbol{e})$ where $u \in \{\max(0, m-t), \ldots, \min(m-1, N-t)\}$ for each t. The resulting vector contains a zero when t bids are equal and there are u bids higher than the tie. The preceding factor $(N+1)$ is large enough to ensure that both addends do not add up to zero. Finally, in the case of a tie, the mth-highest bid can be determined by computing the following additional vectors.

$$\boldsymbol{stat}^m_{t,u}(\boldsymbol{B}) = \Big(\boldsymbol{B} - t\boldsymbol{e} + (N+1)\,(\mathsf{L}\,\boldsymbol{B} - (t+u)\boldsymbol{e})\Big)\mathsf{R}^*$$

Example 2. Suppose that two bids are 50 and two are 20 ($m = 2$, computation takes place in $\mathbb{Z}_{11}$ and $\boldsymbol{p} = (10, 20, 30, 40, 50, 60)$):

$$\boldsymbol{B} = \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 2 \\ 0 \end{pmatrix}$$

$\boldsymbol{stat}^2_{1,1}(\boldsymbol{B})$ yields no information due to the tie at price 50. The first two ($t = 2, u \in \{0, 1\}$) additional order statistic vectors look like this:

$$\boldsymbol{stat}^2_{2,0}(\boldsymbol{B}) = \left(\begin{pmatrix} 0 \\ 2 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} + 5 \left(\begin{pmatrix} 0 \\ 2 \\ 2 \\ 4 \\ 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \right) \right) R^* = \begin{pmatrix} 10 \\ 0 \\ 9 \\ 10 \\ 8 \\ 8 \end{pmatrix} R^* = \begin{pmatrix} * \\ 0 \\ * \\ * \\ * \\ * \end{pmatrix}$$

$$\boldsymbol{stat}^2_{2,1}(\boldsymbol{B}) = \left(\begin{pmatrix} 0 \\ 2 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} + 5 \left(\begin{pmatrix} 0 \\ 2 \\ 2 \\ 4 \\ 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{pmatrix} \right) \right) R^* = \begin{pmatrix} 5 \\ 6 \\ 4 \\ 5 \\ 3 \\ 3 \end{pmatrix} R^* = \begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}$$

For $t > 2$ the first difference contains no zeros, leading to random vectors. The mth-highest bid is indicated in vector $\boldsymbol{stat}^2_{2,0}(\boldsymbol{B})$ (revealing that the two highest bids are equal). ▲

Concluding, in order to obtain the mth order statistic of N bids, agents jointly compute function $\boldsymbol{stat}^m_{t,u}(\boldsymbol{B})$ where $t = \{1, 2, \ldots, N\}$ and $u \in \{\max(0, m - t), \ldots, \min(m - 1, N - t)\}$ for each t. Thus, a total amount of $m(N - m)$ vectors of size k needs to be computed.

5 Multi-Unit Auction Protocols

In this section, we present methods to compute the outcome of three common multi-unit auction types based on the vector notation and the order statistic subprotocol proposed in the previous section.

Before determining the auction outcome, bidders have to prove that their bids are marginal decreasing, *i.e.*, $b^i_m \geq b^i_{m+1}$ for each $m < M$. This can be achieved by computing

$$\boldsymbol{dec}^i_m = \left(\mathsf{L}\, \boldsymbol{b}^i_m + (\mathsf{U} - \mathsf{I})\, \boldsymbol{b}^i_{m+1}\right) \mathsf{R}^i$$

where R^i is a random matrix chosen by bidder i and $\boldsymbol{b}^i_m$ is bidder i's bid vector for the mth unit. Each $\boldsymbol{dec}^i_m$ is jointly decrypted. If any component equals zero, bidder i submitted malformed, *i.e.*, increasing, bids. There is no R_i bidder i could use to hide the fact that one of the components is zero.

As noted in Section 1, the three auction formats only differ in the pricing of units. The *number* of units each bidder receives is identical in all three auction

types. The number of units m^i that bidder i receives can be determined by computing a vector where the component denoting the $(M+1)$st-highest bid is zero and then adding all integrated bid vectors of bidder i. This yields a vector whose components are random except for the single component containing the number of units bidder i receives. In order to squeeze all m^i in the same vector $\boldsymbol{alloc}_{t,u}$, we represent the allocation of units as a base-$(M+1)$ number.[9] Furthermore, bidders jointly compute vector $\boldsymbol{pos}_{t,u}$ that simply indicates the position of the $(M+1)$st-highest bid so that bidders know at which position they find the allocation of units in vector $\boldsymbol{alloc}_{t,u}$.

$$\boldsymbol{pos}_{t,u} = \boldsymbol{stat}_{t,u}^{M+1}\left(\sum_{i=1}^{n}\sum_{m=1}^{M}\boldsymbol{b}_m^i\right)$$

$$\boldsymbol{alloc}_{t,u} = \boldsymbol{stat}_{t,u}^{M+1}\left(\sum_{i=1}^{n}\sum_{m=1}^{M}\boldsymbol{b}_m^i\right) + \mathsf{L}\sum_{i=1}^{n}\left((M+1)^{i-1}\sum_{m=1}^{M}\boldsymbol{b}_m^i\right)$$

Due to certain ties, it is possible that bidders qualify for more units than there are units available. This is the case when there is a tie for the $(M+1)$st-highest bid ($t > 1$) with less than M higher bids ($u < M$). Computing additional vectors that reveal the number of bids each bidder is contributing to the tie allow both bidders and the seller to apply fair (*e.g.*, randomized) methods to select how many units each tied bidder receives.

$$\boldsymbol{surplus}_{t,u} = \boldsymbol{stat}_{t,u}^{M+1}\left(\sum_{i=1}^{n}\sum_{m=1}^{M}\boldsymbol{b}_m^i\right) + \sum_{i=1}^{n}\left((M+1)^{i-1}\sum_{m=1}^{M}\boldsymbol{b}_m^i\right)$$

By computing the above three vectors, bidders are able to determine m^i for each bidder. In the following sections, we show how bidders can privately compute unit prices given by three common multi-unit auction types.

5.1 Uniform-Price Auction

In the uniform-price auction, all bidders pay the same price per unit, given by the $(M+1)$st-highest bid. This information is already contained in vector $\boldsymbol{pos}_{t,u}$. No additional interaction to compute prices is required.

5.2 Discriminatory Auction

In the discriminatory auction bidders pay exactly the sum of amounts they specified in each winning bid. Once, m^i is determined, the price bidder i has to

[9] There are certainly more compact representations, but when assuming that $(M+1)^n$ is less than the size of the underlying finite field, a radix representation has the advantage of being efficiently computable.

pay can be revealed by computing $price^i$ as defined below (please note that this is not a vector).

$$price^i = \sum_{m=1}^{m^i} \sum_{j=1}^{k} j \cdot b^i_{m,j}$$

It is advisable to compute $price^i$ so that only bidder i and the seller get to know it. Other bidders do not need to be informed about the total price bidder i has to pay.

5.3 Generalized Vickrey Auction

The generalized Vickrey auction has the most complex pricing scheme of the auction types we consider. A bidder that receives m^i units pays the sum of the m^i highest losing bids submitted by other bidders, *i.e.*, excluding his own losing bids. Unfortunately, this sophisticated pricing scheme also leads to a higher degree of complexity needed to privately compute Vickrey prices based on our vector representation. The unit prices bidder i has to pay can be determined by invoking the order statistic subprotocol m^i times. In contrast to the discriminatory auction protocol proposed in the previous section, all unit prices have to be computed separately instead of just computing the total price each bidder has to pay. Vector

$$\boldsymbol{price}^i_{m,t,u} = \boldsymbol{stat}^m_{t,u} \left(\sum_{h=1,h\neq i}^{n} \sum_{\ell=m^i+1}^{M} \boldsymbol{b}^h_\ell \right)$$

indicates the price of the mth unit bidder i receives ($m = \{1, 2, \ldots, m^i\}$). Obviously, heavy use of the order statistic protocol results in more information to be revealed in the case of ties. As in the discriminatory auction protocol, unit prices should only be revealed to the seller and corresponding bidders.

6 Implementation Using El Gamal Encryption

Any homomorphic encryption scheme that besides the, say, additive homomorphic operation allows efficient multiplication of encrypted values with a jointly generated random number can be used to implement the schemes described in the previous sections. It turns out that El Gamal encryption [El 85], even though it is multiplicative, is quite suitable because

- agents can easily create distributed keys, and
- encrypted values can be exponentiated with a shared random number in a single round.

As El Gamal cipher is a multiplicative homomorphic encryption scheme, the entire computation as described in the previous sections will be executed in the exponent of a generator. In other words, a random exponentiation implements the random multiplication of the additive notation. As a consequence, the mth-highest bid is marked by ones instead of zeros in the order statistic protocol.

6.1 El Gamal Encryption

El Gamal cipher [El 85] is a probabilistic and homomorphic public-key cryptosystem. Let p and q be large primes so that q divides $p-1$. $\mathbb{G}_q$ denotes $\mathbb{Z}_p^*$'s unique multiplicative subgroup of order q.[10] As argued in Footnote 9, q should be greater than $(M+1)^n$. All computations in the remainder of this paper are modulo p unless otherwise noted. The *private key* is $x \in \mathbb{Z}_q$, the *public key* $y = g^x$ ($g \in \mathbb{G}_q$ is an arbitrary, publicly known element). A message $m \in \mathbb{G}_q$ is *encrypted* by computing the ciphertext tuple $(\alpha, \beta) = (my^r, g^r)$ where r is an arbitrary random number in $\mathbb{Z}_q$, chosen by the encrypter. A message is *decrypted* by computing $\frac{\alpha}{\beta^x} = \frac{my^r}{(g^r)^x} = m$. El Gamal is homomorphic as the componentwise product of two ciphertexts $(\alpha\alpha', \beta\beta') = (mm'y^{r+r'}, g^{r+r'})$ represents an encryption of the plaintexts' product mm'. It has been shown that El Gamal is semantically secure, *i.e.*, it is computationally infeasible to distinguish between the encryptions of any two given messages, if the decisional Diffie-Hellman problem is intractable [TY98].

We will now describe how to apply the El Gamal cryptosystem as a fully private multiparty computation scheme.[11] If a value represents an additive share, this is denoted by a "+" in the index, whereas multiplicative shares are denoted by "×". Underlying zero-knowledge proofs will be presented in the next section.

Distributed key generation: Each agent chooses x_{+i} at random and publishes $y_{\times i} = g^{x_{+i}}$ along with a zero-knowledge proof of knowledge of $y_{\times i}$'s discrete logarithm. The public key is $y = \prod_{i=1}^{n} y_{\times i}$, the private key is $x = \sum_{i=1}^{n} x_{+i}$. The broadcast round complexity and the computational complexity of the key generation are $\mathcal{O}(1)$.

Distributed decryption: Given an encrypted message (α, β), each agent publishes $\beta_{\times i} = \beta^{x_{+i}}$ and proves its correctness. The plaintext can be derived by computing $\frac{\alpha}{\prod_{i=1}^{n} \beta_{\times i}}$. Like the key generation, the decryption can be performed in constant time.

Random Exponentiation: A given encrypted value (α, β) can easily be raised to the power of an unknown random number $E = \sum_{i=1}^{n} e_{+i}$ whose addends can be freely chosen by the agents if each bidder publishes $(\alpha^{e_{+i}}, \beta^{e_{+i}})$ and proves the equality of logarithms. The product of published ciphertexts yields (α^E, β^E) in a single step.

6.2 Zero-Knowledge Proofs

In order to obtain security against *malicious* or so-called *active* adversaries, bidders are required to prove the correctness of each protocol step. One of the objectives when designing the protocols presented in Section 5 was to enable *efficient*

[10] We will focus on multiplicative subgroups of finite fields here, although El Gamal can also be based on other groups such as elliptic curve groups.

[11] Please note that this multiparty scheme is limited in the sense that it does not allow the computation of *arbitrary* functions.

proofs of correctness for protocol steps. In fact, the proposed protocols can be proven correct by only using so-called Σ-protocols which just need three rounds of interaction [Dam02,CDS94]. Σ-protocols are not known to be zero-knowledge, but they satisfy the weaker property of *honest-verifier* zero-knowledge. This suffices for our purposes as we can use the Fiat-Shamir heuristic [FS87] to make these proofs non-interactive. As a consequence, the obtained proofs are indeed zero-knowledge *in the random oracle model* and only consist of a single round.[12] We will make use of the following three Σ-protocols:

- Proof of knowledge of a discrete logarithm [Sch91]
- Proof of equality of two discrete logarithms [CP92]
- Proof that an encrypted value is one out of two values [CGS97]

6.3 Protocol Implementation

Using El Gamal encryption, the computation schemes described in Section 5 can be executed in the exponent of an arbitrary value in $\mathbb{G}_q \backslash \{1\}$ that is known to all bidders. When enabling non-interactive zero-knowledge proofs by applying the Fiat-Shamir heuristic (see Section 6.2), protocols only require a low constant number of rounds of broadcasting.[13] The allocation of units can be computed in four rounds as described below. Additional rounds may be required to compute unit prices depending on the auction type.

- Round 1: Distributed generation of El Gamal keys.
- Round 2: Publishing El Gamal encryptions of bids and proving their correctness.
- Round 3: Joint computation of $\boldsymbol{pos}_{t,u}$, $\boldsymbol{alloc}_{t,u}$, and $\boldsymbol{surplus}_{t,u}$ as defined in Section 5. One round of interaction is needed for random exponentiation.
- Round 4: Distributed decryption of $\boldsymbol{pos}_{t,u}$, $\boldsymbol{alloc}_{t,u}$, and $\boldsymbol{surplus}_{t,u}$.

These four rounds suffice to determine the outcome of the uniform-price auction. The discriminatory auction requires one additional round of interaction for computing $price^i$. This cannot be integrated in round 3 because m^i needs to be known for computing $price^i$. The generalized Vickrey auction requires two additional rounds due to random exponentiations needed for computing $\boldsymbol{price}^i_{m,t,u}$.

In round 4, bidders send decrypted shares of the outcome to the seller rather than publishing them immediately. After the seller received all shares, he publishes them. This ensures that no bidder can quit the protocol prematurely after learning the outcome, thus leaving other bidders uninformed (see also [Bra03a]). The same procedure is applied in round 5 or 6, respectively, with the difference that the seller does not need to publish shares. As mentioned in Sections 5.2 and 5.3, it suffices to send information on unit prices to the corresponding bidder.

[12] The additional assumption of a random oracle is only made for reasons of efficiency. Alternatively, we could employ non-interactive zero-knowledge proofs in the *common random string model* (see [DDO+01] and references therein). However, it has become common practice to use secure hash functions like MD5 or SHA-1 as random oracles in practice.

[13] As explained in Section 2, we do not consider the additional overhead caused by bidders that abort the protocol.

7 Conclusion

We proposed general cryptographic protocols for three common types of multi-unit auctions based on distributed homomorphic encryption and concrete implementations of these protocols using El Gamal cipher. The security of El Gamal encryption as well as the applied zero-knowledge proofs can be based on the decisional Diffie-Hellman assumption. Under this assumption, privacy can not be breached (unless *all* bidders collude). Our protocols reveal the following information if there is a tie at the $(M+1)$st-highest bid: the number of tied bids (t) and the number of bids greater than the tie (u). The generalized Vickrey auction protocol additionally reveals the price of each unit (rather than just the summed up prices each bidder has to pay) and related tie information. Protocols only fail when the random exponentiation "accidently" yields a one. Due to the exponential size of $\mathbb{G}_q$ the probability of this event is negligible.

In the discriminatory and generalized Vickrey auction protocol, sanctions or fines need to be imposed on bidders that quit prematurely because the allocation and the prices of units are revealed in two consecutive steps. A bidder that learns that he will not receive a single unit might decide to quit the protocol. However, his continuing participation is required to compute the prices of units.

Auction Type	# of Rounds	Exponentiations
Uniform-Price	4	$\mathcal{O}(nM^2k)$
Discriminatory	5	$\mathcal{O}(nM^2k)$
Generalized Vickrey	6	$\mathcal{O}(nM^3k)$

n: bidders, k: prices/possible bids, M: units to be sold

Table 1. Protocol Complexity (Computation per Bidder)

Table 1 shows the complexity of the proposed protocols (in the random oracle model). Round complexity is very low, but communication and computation complexity is linear in k (rather than logarithmic when using binary representations of bids). On the other hand, an advantage of the unary vector representation is that protocols can easily be turned into iterative auction protocols.

Acknowledgements

This material is based upon work supported by the Deutsche Forschungsgemeinschaft under grant BR 2312/1-1, by the National Science Foundation under grants IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678, and a Sloan Fellowship.

References

[AS02] M. Abe and K. Suzuki. M+1-st price auction using homomorphic encryption. In *Proc. of 5th International Conference on Public Key Cryptography (PKC)*, volume 2274 of *LNCS*, pages 115–224. Springer, 2002.

[Bra02] F. Brandt. Secure and private auctions without auctioneers. Technical Report FKI-245-02, Technical University of Munich, 2002. ISSN 0941-6358.

[Bra03a] F. Brandt. Fully private auctions in a constant number of rounds. In R. N. Wright, editor, *Proc. of 7th FC Conference*, volume 2742 of *LNCS*, pages 223–238. Springer, 2003.

[Bra03b] F. Brandt. Social choice and preference protection - Towards fully private mechanism design. In N. Nisan, editor, *Proc. of 4th ACM Conference on Electronic Commerce*, pages 220–221. ACM Press, 2003.

[BS01] O. Baudron and J. Stern. Non-interactive private auctions. In *Proc. of 5th FC Conference*, pages 300–313, 2001.

[BS04a] F. Brandt and T. Sandholm. (Im)possibility of unconditionally privacy-preserving auctions. In C. Sierra and L. Sonenberg, editors, *Proc. of 3rd AAMAS Conference*, pages 810–817. ACM Press, 2004.

[BS04b] F. Brandt and T. Sandholm. On correctness and privacy in distributed mechanisms. In P. Faratin and J. A. Rodriguez-Aguilar, editors, *Selected and revised papers from the 6th AAMAS Workshop on Agent-Mediated Electronic Commerce (AMEC)*, LNAI, pages 1–14, 2004.

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proc. of 14th CRYPTO Conference*, volume 893 of *LNCS*, pages 174–187. Springer, 1994.

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Proc. of 14th Eurocrypt Conference*, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.

[Cla71] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[CP92] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Proc. of 12th CRYPTO Conference*, volume 740 of *LNCS*, pages 3.1–3.6. Springer, 1992.

[CSS05] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2005. To appear.

[Dam02] I. Damgård. On Σ-protocols. Lecture Notes, University of Aarhus, Department for Computer Science, 2002.

[DDO+01] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In *Proc. of 21th CRYPTO Conference*, volume 2139 of *LNCS*, pages 566–598. Springer, 2001.

[Di 00] G. Di Crescenzo. Private selective payment protocols. In *Proc. of 4th FC Conference*, volume 1962 of *LNCS*. Springer, 2000.

[El 85] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proc. of 12th CRYPTO Conference*, LNCS, pages 186–194. Springer, 1987.

[GL90] S. Goldwasser and L. Levin. Fair computation of general functions in presence of immoral majority. In *Proc. of 10th CRYPTO Conference*, volume 537 of *LNCS*, pages 77–93. Springer, 1990.

[GL02] S. Goldwasser and Y. Lindell. Secure computation without agreement. In *Proc. of 16th International Symposium on Distributed Computing (DISC)*, volume 2508 of *LNCS*, pages 17–32. Springer, 2002.

[Gro73] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[Gro04] J. Groth. Efficient maximal privacy in boardroom voting and anonymous broadcast. In *Proc. of 8th FC Conference*, volume 3110 of *LNCS*, pages 90–104. Springer, 2004.

[JS02] A. Juels and M. Szydlo. A two-server, sealed-bid auction protocol. In M. Blaze, editor, *Proc. of 6th FC Conference*, volume 2357 of *LNCS*. Springer, 2002.

[Kik01] H. Kikuchi. (M+1)st-price auction protocol. In *Proc. of 5th FC Conference*, volume 2339 of *LNCS*, pages 351–363. Springer, 2001.

[Kle99] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.

[Kri02] V. Krishna. *Auction Theory*. Academic Press, 2002.

[KY02] A. Kiayias and M. Yung. Self-tallying elections and perfect ballot secrecy. In *Proc. of 5th PKC Conference*, number 2274 in LNCS, pages 141–158. Springer, 2002.

[KY03] A. Kiayias and M. Yung. Non-interactive zero-sharing with applications to private distributed decision making. In *Proc. of 7th FC Conference*, volume 2742 of *LNCS*, pages 303–320. Springer, 2003.

[LAN02] H. Lipmaa, N. Asokan, and V. Niemi. Secure Vickrey auctions without threshold trust. In M. Blaze, editor, *Proc. of 6th FC Conference*, volume 2357 of *LNCS*. Springer, 2002.

[NPS99] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proc. of 1st ACM Conference on E-Commerce*, pages 129–139. ACM Press, 1999.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[SS01] T. Sandholm and S. Suri. Market clearability. In *Proc. of 17th IJCAI*, pages 1145–1151, 2001.

[SY02] K. Suzuki and M. Yokoo. Secure combinatorial auctions by dynamic programming with polynomial secret sharing. In M. Blaze, editor, *Proc. of 6th FC Conference*, volume 2357 of *LNCS*. Springer, 2002. to appear.

[SY03] K. Suzuki and M. Yokoo. Secure generalized Vickrey auction using homomorphic encryption. In *Proc. of 7th FC Conference*, volume 2742 of *LNCS*, pages 239–249. Springer, 2003.

[Ten00] M. Tennenholtz. Some tractable combinatorial auctions. In *Proc. of 17th AAAI Conference*, pages 98–103. AAAI Press / The MIT Press, 2000.

[TY98] Y. Tsiounis and M. Yung. On the security of ElGamal-based encryption. In *Proc. of 1st International Workshop on Practice and Theory in Public Key Cryptography (PKC)*, volume 1431 of *LNCS*, pages 117–134. Springer, 1998.

[Vic61] W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[Yao86] A. C. Yao. How to generate and exchange secrets. In *Proc. of 27th FOCS Symposium*, pages 162–167. IEEE Computer Society Press, 1986.

[YS04] M. Yokoo and K. Suzuki. Secure generalized Vickrey auction without third-party servers. In *Proc. of 8th FC Conference*, volume 3110 of *LNCS*. Springer, 2004.