Fully Private Auctions in a Constant Number of Rounds

Revised FC 2003 Paper (Feb. 2004)*

Felix Brandt

Computer Science Department
Technical University of Munich
brandtf@cs.tum.edu

Abstract. We present a new cryptographic auction protocol that prevents extraction of bid information despite any collusion of participants. This requirement is stronger than common assumptions in existing protocols that prohibit the collusion of certain third-parties (e.g. distinct auctioneers). Full privacy is obtained by using homomorphic ElGamal encryption and a private key that is distributed among the set of bidders. Bidders jointly compute the auction outcome on their own without uncovering any additional information in a constant number of rounds (three in the random oracle model). No auctioneers or other trusted third parties are needed to resolve the auction. Yet, robustness is assured due to public verifiability of the entire protocol. The scheme can be applied to any uniform-price (or so-called $(M+1)$st-price) auction. An additional, optional, feature of the protocol is that the selling price is only revealed to the seller and the winning bidders themselves. We furthermore provide an in-depth analysis of ties in our protocol and sketch a scheme that requires more rounds but is computationally much more efficient.

1 Introduction

Auctions have become the major phenomenon of electronic commerce during the last years. In recent times, the need for privacy has been a factor of increasing importance in auction design and various schemes to ensure the safe conduction of sealed-bid auctions have been proposed.

We consider a situation where one seller and n bidders or buyers intend to come to an agreement on the selling of a good[1]. Each bidder submits a sealed bid expressing how much he is willing to pay. The bidders want the highest bidder to win the auction for a price that has to be determined by a publicly known rule (e.g. the highest or second-highest bid). In order to fulfill this task, they need a trusted third-party, which is called the "auctioneer". Among the different auction protocols, the second-price or so-called Vickrey auction [1], where the highest bidder wins by paying the amount of the second-highest bid,

* Thanks to Jens Groth for pointing out some errors in Section 5.1.

[1] The assignment of tasks in reverse auctions works similarly.

has received particular attention in recent times because it is "strategy-proof", i.e., bidders are always best off bidding their private valuation of a good. This is a huge advantage over first-price auctions, where bidders have to estimate the other bidders' valuations when calculating their bid. However, despite its impressive theoretical properties, the Vickrey auction is rarely used in practice. It is generally agreed [2–4] that the Vickrey auction's sparseness is due to two major reasons: the *fear of an untruthful auctioneer* and the *reluctance of bidders to reveal their true valuations*. The winner of an auction has to doubt whether the price the auctioneer tells him to pay is actually the second-highest bid. The auctioneer could easily make up a "second-highest" bid to increase his (or the seller's) revenue. In addition to a possibly insincere auctioneer, bidders have to reveal their valuations to the auctioneer. There are numerous ways to misuse these values by giving them away to other bidders or the seller [5–7]. It remains in the hands of the auctioneer whether the auction really is a *sealed*-bid auction.

The proposed protocol removes both crucial weaknesses of the Vickrey auction by omitting the auctioneer and distributing the calculation of the selling price on the bidders themselves. No information concerning the bids is revealed unless *all* bidders share their knowledge, which obviously uncovers all bids in any auction protocol. Furthermore, our protocol is applicable to a generalization of the Vickrey auction called uniform-price or $(M+1)$st-price auction. In an $(M+1)$st-price auction, the seller offers M identical items and each bidder intends to buy *one* of them. It has been proven that it is a strategy-proof mechanism to sell those items to the M highest bidders for the uniform price given by the $(M+1)$st highest bid [1, 8]. The Vickrey auction is just a special case of this mechanism for the selling of single goods ($M = 1$).

Thus, our main contribution is a verifiable protocol for n participants, each having a secret value, that only reveals the $(M+1)$st highest value to the M participants who possess higher values.

The remainder of this paper is structured as follows. Section 2 summarizes existing efforts in the field of cryptographic auction protocols. Section 3 defines essential attributes that ensure a secure and private auction conduction and Section 4 introduces "bidder-resolved auctions". In Section 5, we propose a bidder-resolved $(M+1)$st-price auction protocol. The paper concludes with an overview of the protocol's complexity and a brief outlook in Section 6.

2 Related Work

There has been a very fast-growing interest in cryptographic protocols for auctions during the last years. In particular, Vickrey auctions and recently the more general $(M+1)$st-price auctions attracted much attention. Starting with the work by Franklin and Reiter [9], which introduced the basic problems of sealed-bid auctions, but disregarded the privacy of bids after the auction is finished, many secure auction mechanisms have been proposed, e.g. [7, 10–26].

When taking away all the protocols that (in their current form) are only suitable for the secure execution of *first*-price auctions or reveal (partial) information

after the auction is finished [7, 9, 11, 15, 19, 22, 23, 27, 25, 26], the remaining work can be divided into two categories.

Most of the publications rely on threshold computation that is distributed among auctioneers [14, 16–18, 24]. This technique requires m auctioneers, out of which a fraction (mostly a majority) must be trustworthy. Bidders send shares of their bids to each auctioneer. The auctioneers jointly compute the selling price without ever knowing a single bid. This is achieved by using techniques like verifiable secret sharing and secure multiparty function evaluation. However, a collusion of, e.g., three out of five auctioneer servers can already exploit the bidders' trust. We argue that distributing the trust onto several distinct auctioneers does not solve the privacy problem, because you can never rule out that some of them, or even *all* of them, collude.

The remaining auction protocols prune the auctioneer's ability to falsify the auction outcome and reveal confidential information by introducing a new third-party that is not *fully* trusted. However, all of these approaches make weak assumptions about the trustworthiness of this third-party. In [12, 13] the third-party may not collude with any participating bidder; in [20, 21] it is prohibited that the third-party and the auctioneer collude. A recent scheme [10] uses a homomorphic, indistinguishable public-key encryption scheme like ElGamal to compute on encrypted bids. However, the private key is either held by a trusted third-party or is shared among a set of confidants which makes the protocol as safe as the ones using several auctioneers (see Section 4 for information on how this scheme can be distributed on bidders).

Concluding, all present work on secure auctions more or less relies on the exclusion of third-party collusion, may it be auctioneers or other semi-trusted institutions. Additionally, many of the existing schemes publicly announce the winner's identity and all of them declare the selling price rather than making this information only visible to the seller and the winners.

3 General Assumptions

This section specifies demands that a secure auction protocol has to meet. Furthermore, we make several rigorous assumptions about auction participants and collusions between them. The required properties for safe conductions of sealed-bid auctions can be divided into two categories.

Privacy *No information concerning bids and the corresponding bidders' identities is revealed during and after the auction.*

The only information that naturally has to be delivered is the information that is needed to carry out the transaction, i.e., the winning bidders and the seller learn the selling price and the seller gets to know the winners' identities. As [27] pointed out, *anonymity* of the winners is crucial. Otherwise, a bidder that breaks a collusive agreement could be identified by his partners. [11] introduced the property of "receipt-freeness" in the context of auctions.

It prevents bidders from proving their bidding prices in order to circumvent bid-rigging. Our protocol is not receipt-free as this would heavily affect efficiency and because receipt-freeness requires untappable channels.

Privacy, as we understand it, implies that no information on any bid is revealed to the public, in particular no bid statistics (e.g. the amount of the lowest bid or an upper bound for the highest bid) can be extracted.

Correctness *The winner and the selling price are determined correctly.*
This requirement includes *non-repudiation* (winning bidders cannot deny having made winning bids) and the immutability of bids. *Robustness* (no subset of malicious bidders can render the auction outcome invalid) also belongs to this category (see Section 4).

Privacy and correctness have to be ensured in a hostile environment as we allow every feasible type of collusion. We assume that up to $n-1$ bidders might share their knowledge and act as a team. This implies that each bidder can have arbitrarily many bidder sub-agents, controlled by him. Besides, the seller might collude with bidders, and any number of auctioneers or other third parties might collude and are therefore not trustworthy. We assume the standard model of a secure broadcast channel.

4 Bidder-resolved Auctions

According to the assumptions of the previous section, bidders cannot trust any third-party. We therefore distribute the trust onto the bidders themselves. This allows us to set a new standard for privacy. In a scenario with m auctioneers it cannot be ruled out that all of them collude. However, when distributing the computation on n bidders, we can assume that *all* bidders will never share their knowledge due to the competition between them. If they did, each of them would completely abandon his own privacy, resulting in a public auction. We therefore argue, that only bidder-resolved auctions provide *full privacy*, i.e., no information on any bid can be retrieved unless all bidders collude. Full privacy can be interpreted as $(n-1)$-privacy or (n,n)-threshold privacy.

It is difficult to assure robustness in bidder-resolved auctions. However, verifiability can be used to provide what we call *weak robustness*, so that malicious bidders will be detected immediately (without additional communication and information revelation) and can be excluded from the set of bidders. The protocol can then be restarted with the remaining bidders proving that their bids did not change[2]. This guarantees termination (after at most $n-M$ iterations) and correctness (if we agree that the removal of malicious bidders (and their bids) does not violate correctness). As malicious bidders can easily be fined and they do not gain any information, there should be no incentive to perturb the auction and we henceforth assume that a single protocol run suffices.

[2] This is not mandatory as their should be no reason to strategically change a bid after a bidder has been excluded (assuming the private-value model).

Public verifiability of the protocol is sufficient to provide weak robustness and verifiability can be easily achieved by using zero-knowledge proofs. Unfortunately, when abandoning (strong) robustness, we also lose "fairness". Typically, in the end of a protocol run, each participant holds a share of the result. As simultaneous publication of these shares is impossible, a malicious agent might quit the protocol after having learned the result but before others were able to learn it. There are various techniques to approximate fairness by gradually releasing parts of the secrets to be swapped. Another possibility is to introduce a third-party that publishes the outcome after it received all shares. This third-party does not learn confidential information. It is only assumed not to leave the protocol prematurely. We believe that in auctions with a single seller, it is practical to assign this role to the seller. This obviously leaves the possibility of a "cheating seller" who quits the protocol after having learned the (possibly unsatisfying) result. However, such a seller could be forced to sell the good for the resulting price as bidders can compute the auction outcome on their own (or with another fairness-providing third party).

The naive approach to build a Boolean circuit that computes the auction outcome on binary representations of bids by applying a general multiparty computation (MPC) scheme is not feasible as those schemes are quite inefficient and the circuit depth, and thus the round complexity, depends on the number of bidders and the bid size. Like many other existing schemes, we therefore use an ordered set of k possible prices (or valuations) $(p_1, p_2, \ldots, p_k)$. This results in linear computational complexity but enables special purpose protocols that do not require a general MPC scheme. In fact, our protocol has constant round complexity because it only uses additions and no multiplications.

A framework for bidder-resolved auction protocols could look like this:

- The seller publicly announces the selling of a certain good by publishing
  - the good's description,
  - the amount of units to be sold,
  - the registration deadline,
  - lower and upper bounds of the valuation interval, and
  - a function that prescribes how and how many valuations $(p_1, p_2, \ldots, p_k)$ are distributed among that interval subject to the number of bidders n (enabling linear, logarithmic, or any other form of scaling)

  on a blackboard.
- Interested bidders publish their id's on the blackboard.
  — *registration deadline* —
- The bidders jointly compute the winners and the selling price.

A threshold-scheme providing t-resilience is not appropriate when information is shared among bidders, as any group of bidders might collude due to the assumptions of the previous section. As a consequence, we cannot simply adapt existing auction protocols that were designed for multiple auctioneers. Protocols like [14], or [16] rely on information-theoretic secure multiparty computation according to Ben-Or, Goldwasser and Wigderson [28], which provides at most

insufficient $\lfloor \frac{n-1}{2} \rfloor$-privacy due to the multiplication of degree n polynomials.
Another recent protocol by Abe and Suzuki [10] uses verifiable mix and match [15] of ElGamal ciphertexts assuming an honest majority due to robustness requirements. When relaxing these requirements in order to realize a bidder-resolved protocol and discarding binary search to minimize the round complexity, mixing would still require $\mathcal{O}(n)$ rounds. With further changes that enable privacy of the selling price, the computational and message complexity per bidder would be $\mathcal{O}(nkM \log(M))$, which is fairly good as M is negligibly small in many cases (e.g. in a Vickrey auction). However, the number of rounds depends on the number of bidders n.

5 Protocol Description

We will use an additive vector notation to describe our approach. The actual implementation described in Section 5.1, however, will take place in a multiplicative group using ElGamal encryption with a public key that is jointly created by all bidders.
Each bidder sets the bid vector[3]

$$\boldsymbol{b}_i = (b_{i1}, b_{i2}, \ldots, b_{ik}) = (\underbrace{0, \ldots, 0}_{b_i - 1}, Y, \underbrace{0, \ldots, 0}_{k - b_i})$$

according to his bid $b_i \in \{1, 2, \ldots, k\}$, publishes its encryption, and shows its correctness by proving $\forall j \in \{1, 2, \ldots, k\} : \; b_{ij} \in \{0, Y\}$ and $\sum_{j=1}^{k} b_{ij} = Y$ in zero-knowledge manner (like in [10]). $Y \neq 0$ is a generally known group element, e.g. 1.

The homomorphic encryption scheme allows verifiable computation of linear combinations of secrets in a single round. When computing on vectors of homomorphically encrypted values (like $\boldsymbol{b}_i$), this means that besides addition and substraction of (encrypted) vectors, multiplication with (known) matrices is feasible. For example, the "integrated"[10] bid vector

$$\boldsymbol{b}'_i = (\underbrace{Y, \ldots, Y}_{b_i}, \underbrace{0, \ldots, 0}_{k - b_i}) = (b_{i1} + b'_{i2}, b_{i2} + b'_{i3}, \ldots, b_{ik})$$

can be derived by multiplying the bid vector with the $k \times k$ lower triangular matrix L ($\boldsymbol{b}'_i = \mathsf{L}\boldsymbol{b}_i$).

$$\mathsf{L} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix} \qquad \text{(lower triangular matrix)}$$

[3] To save space, vector components are listed horizontally (bottom-up).

Multiplying a vector with $\mathsf{L}-\mathsf{I}$, where I is the $k \times k$ identity matrix, yields $\boldsymbol{b}'_i$ shifted down by one component.

$$\mathsf{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \qquad \text{(identity matrix)}$$

If we sum up all integrated bid vectors and down-shifted integrated bid vectors, we obtain a vector that has the following structure (let us for now disregard the possibility of equal bids, we will refer to this case in Section 5.2).

$$(2\mathsf{L}-\mathsf{I})\sum_{i=1}^{n}\boldsymbol{b}_i = (\ldots, 6Y, \ldots, 6Y, 5Y, 4Y, \ldots, 4Y, 3Y, 2Y, \ldots, 2Y, Y, 0, \ldots, 0)$$

The position of the (single) component that equals $3Y$ denotes the second-highest bid, $5Y$ the third-highest bid, and so forth. Subtracting $(2M+1)Y\boldsymbol{e}$ with $\boldsymbol{e} = (1, \ldots, 1)$, thus yields a vector in which the component, that refers to the amount of the $(M+1)$st highest bid, is 0. All other components are not 0.
As we intend to create personal indicators for each bidder, we mask the resulting vector so that only winning bidders can read the selling price. This is achieved by adding $\mathsf{U}\boldsymbol{b}_i$.

$$\mathsf{U} = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \qquad \text{(upper triangular matrix)}$$

For an arbitrary bidder a, the vector $(2\mathsf{L}-\mathsf{I})\sum_{i=1}^{n}\boldsymbol{b}_i - (2M+1)Y\boldsymbol{e} + (2M+2)\mathsf{U}\boldsymbol{b}_a$ only contains a component equal 0, when a qualifies as a winner of the auction. The position of this component then indicates the selling price.
In order to get rid of all information besides the selling price, each component is multiplied with a different random multiplier M_{ij} that is jointly created and unknown to any subset of bidders. Finally, each bidder's personal indicator vector is computed according to the following equation.

$$\boldsymbol{v}_a = \left((2\mathsf{L}-\mathsf{I})\sum_{i=1}^{n}\boldsymbol{b}_i - (2M+1)Y\boldsymbol{e} + (2M+2)\mathsf{U}\boldsymbol{b}_a\right)\mathsf{R}^*_a$$

$$\mathsf{R}^*_i = \begin{pmatrix} M_{ik} & 0 & \cdots & 0 \\ 0 & M_{i,k-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & M_{i1} \end{pmatrix} \qquad \text{(random multiplication matrix)}$$

Be aware that R_i^* does *not* represent a feasible linear operation on encrypted values as the homomorphic property only provides addition, but not multiplication, of secrets. The components M_{ij} are unknown to bidders. In Section 5.1, we will present a very efficient way to randomize ElGamal encrypted vector components.

The invariant of the "blinding" transformation are components that equal 0. As described before, those components mark the selling price to winning bidders. Only bidder i and the seller get to know $\boldsymbol{v}_i$.

$$v_{ij} = 0 \quad \Longleftrightarrow \quad \text{Bidder } i \text{ won and has to pay } p_j$$

The following simple example for two bidders illustrates the functionality of the protocol. The computations take place in $\mathbb{Z}_{11}$ and the auction to be conducted is a Vickrey auction ($M = 1$). Bids are $b_1 = 2$ and $b_2 = 5$: $\boldsymbol{b}_1 = (0,1,0,0,0,0)$, $\boldsymbol{b}_2 = (0,0,0,0,1,0)$. The selling price can be determined by computing

$$\begin{pmatrix} 1&0&0&0&0&0\\ 2&1&0&0&0&0\\ 2&2&1&0&0&0\\ 2&2&2&1&0&0\\ 2&2&2&2&1&0\\ 2&2&2&2&2&1 \end{pmatrix} \left(\begin{pmatrix} 0\\0\\0\\0\\1\\0 \end{pmatrix} + \begin{pmatrix} 0\\1\\0\\0\\0\\0 \end{pmatrix} \right) - \begin{pmatrix} 3\\3\\3\\3\\3\\3 \end{pmatrix} = \begin{pmatrix} 0\\1\\2\\2\\3\\4 \end{pmatrix} - \begin{pmatrix} 3\\3\\3\\3\\3\\3 \end{pmatrix} = \begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} \quad .$$

Now, the selling price has to be masked to losing bidders. Bidder 1 is unable to identify the selling price. His indication vector ($\boldsymbol{v}_1$) contains random numbers.

$$\begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + 4 \begin{pmatrix} 1&1&1&1&1&1\\ 0&1&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&0&1&1&1\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 0\\0\\0\\0\\1\\0 \end{pmatrix} = \begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + \begin{pmatrix} 4\\4\\4\\4\\4\\0 \end{pmatrix} = \begin{pmatrix} 1\\2\\3\\3\\4\\1 \end{pmatrix} \overset{\times R_1^*}{\longrightarrow} \begin{pmatrix} .\\.\\.\\.\\.\\. \end{pmatrix}$$

Bidder 2's indicator $\boldsymbol{v}_2$, however, indicates the selling price at the second component (bottom-up).

$$\begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + 4 \begin{pmatrix} 1&1&1&1&1&1\\ 0&1&1&1&1&1\\ 0&0&1&1&1&1\\ 0&0&0&1&1&1\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 0\\1\\0\\0\\0\\0 \end{pmatrix} = \begin{pmatrix} 8\\9\\10\\10\\0\\1 \end{pmatrix} + \begin{pmatrix} 4\\4\\0\\0\\0\\0 \end{pmatrix} = \begin{pmatrix} 1\\2\\10\\10\\0\\1 \end{pmatrix} \overset{\times R_2^*}{\longrightarrow} \begin{pmatrix} .\\.\\.\\.\\0\\. \end{pmatrix}$$

5.1 Implementation using ElGamal Encryption

The following implementation of the protocol is based on a distributed version of ElGamal cipher and uses several zero-knowledge proofs like Schnorr's proof of knowledge of a discrete logarithm [29], Chaum and Pedersen's proof of equality of discrete logarithms [30], and Cramer, Damgård, and Schoenmaker's proof of partial knowledge [31]. It is much more efficient than a previous version [32] that was based on verifiable secret sharing. Indices $+i$ and $\times i$ are used to indicate additive and multiplicative shares, respectively.

ElGamal cipher [33] is a probabilistic public-key cryptosystem that provides two very useful properties: homomorphic and semantically secure encryption [34]. p and q are large primes so that q divides $p-1$. $\mathbb{G}_q$ denotes $\mathbb{Z}_p$'s unique multiplicative subgroup of order q. The private key is $x \in \mathbb{Z}_q$, the public key $y = g^x$ ($g \in \mathbb{G}_q$). 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 number in $\mathbb{Z}_q$. A message is decrypted by computing $\frac{\alpha}{\beta^x} = \frac{my^r}{(g^r)^x} = m$. The product of two ciphertexts $(\alpha\alpha', \beta\beta')$ represents an encryption of the plaintexts' product mm' (homomorphic property). We will now describe how to apply the ElGamal cryptosystem as a fully private, i.e. non-threshold, multiparty computation scheme.

Distributed key generation: Each participant 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 using [29]. 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 participant publishes $\beta_{\times i} = \beta^{x_{+i}}$ and proves its correctness (as described in [30]). 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 $M = \sum_{i=1}^{n} m_{+i}$ whose addends can be freely chosen by the participants if each bidder publishes $(\alpha^{m_{+i}}, \beta^{m_{+i}})$ and proves the equality of logarithms. The product of the published ciphertexts yields (α^M, β^M). Random Exponentiation can thus be executed in a single step. Random exponentiation was the bottleneck of our previous auction protocol [32] that was based on verifiable secret sharing.

What follows is the step-by-step protocol specification for bidder a and his bid b_a. $i, h \in \{1, 2, \ldots, n\}$, and $j, b_a \in \{1, 2, \ldots, k\}$. $Y \in \mathbb{G}_q \backslash \{1\}$ is known to all bidders.

1. Choose x_{+a} and $\forall i, j:\ m_{ij}^{+a}, r_{aj} \in \mathbb{Z}_q^*$ at random.
2. Publish $y_{\times a} = g^{x_{+a}}$ along with a zero-knowledge proof of knowledge of $y_{\times a}$'s discrete logarithm using [29].
3. Compute $y = \prod_{i=1}^{n} y_{\times i}$.
4. $\forall j:$ Set $b_{aj} = \begin{cases} Y & \text{if } j = b_a \\ 1 & \text{else} \end{cases}$ and publish $\alpha_{aj} = b_{aj} y^{r_{aj}}$ and $\beta_{aj} = g^{r_{aj}}$.

5. Prove that $\forall j:\ \log_g(\beta_{aj})$ equals either $\log_y(\alpha_{aj})$ or $\log_y\left(\frac{\alpha_{aj}}{Y}\right)$ ([31]), and that $\log_y\left(\frac{\prod_{j=1}^{k}\alpha_{aj}}{Y}\right)=\log_g\left(\prod_{j=1}^{k}\beta_{aj}\right)$ ([30]).
6. Compute[4] $\forall i,j:\ \gamma_{ij}=\frac{\prod_{h=1}^{n}\prod_{d=j}^{k}(\alpha_{hd}\alpha_{h,d+1})\left(\prod_{d=1}^{j}\alpha_{id}\right)^{2M+2}}{Y^{2M+1}}$ and $\delta_{ij}=\prod_{h=1}^{n}\prod_{d=j}^{k}(\beta_{hd}\beta_{h,d+1})\left(\prod_{d=1}^{j}\beta_{id}\right)^{2M+2}$.
7. Publish $\forall i,j:\ \gamma_{ij}^{\times a}=(\gamma_{ij})^{m_{ij}^{+a}}$ and $\delta_{ij}^{\times a}=(\delta_{ij})^{m_{ij}^{+a}}$ with a proof of their correctness ([30]).
8. Send $\forall i,j:\ \varphi_{ij}^{\times a}=(\prod_{h=1}^{n}\delta_{ij}^{\times h})^{x_{+a}}$ with a proof of using the same x_{+a} as in step 2 ([30]) to the seller who publishes all $\varphi_{ij}^{\times h}$ and the corresponding proofs of correctness for each $i,j,h\neq i$ after having received all of them.
9. Compute $v_{aj}=\frac{\prod_{i=1}^{n}\gamma_{aj}^{\times i}}{\prod_{i=1}^{n}\varphi_{aj}^{\times i}}$.
10. If $v_{aw}=1$ for any w, then bidder a is a winner of the auction. p_w is the selling price.

The final steps are conducted in a way that allows the seller to assemble all decrypted indicators before the bidders can compute them. This prevents a winning bidder from aborting the protocol after having learned the auction result. Alternatively, a sub-protocol that enables "fair exchange of secrets" could be used while including the seller into the secret sharing process.
Assuming the random oracle model that allows non-interactive zero-knowledge proofs, the protocol requires just three rounds of interaction (four rounds including joint key generation).

5.2 The Problem of Equal Bids

When two or more bidders have the $(M+1)$st highest bid in common, the protocol yields no winners. There is no information revelation in this case, except that there has been a tie on the $(M+1)$st highest bid. However, this might be used by a group of malicious bidders who submit equal bids on purpose to learn about the selling price. If the tie is undetected, their bids were lower than the selling price. If the protocol fails, their bids were at least as high as the selling price would have been (without their participation). Besides, ties can be used to destroy the protocol's robustness, as tieing bidders can anonymously disrupt the auction. In the following, we will discuss three different methods to circumvent the tie problem. The first two avoid ties while the last one identifies ties.

[4] $\alpha_{h,k+1}$ and $\beta_{h,k+1}$ are defined as 1 for any h.

"Interlacing" Vector Components (INT) A straight-forward way to avoid the problem is to increase the number of components in $\boldsymbol{v}_i$ from k to nk and insert bidder i's bid in row $nj+i-1$. This increases the computational complexity to $\mathcal{O}(n^2k)$. Unfortunately, this method reveals the identity of one of the $(M+1)$st highest bidders to the winners.

Preventing Equal Bids (PRE) Exact bid amounts b_i can be computed by summing up the components of $\mathsf{L}\boldsymbol{b}_i$. The equality of bids can be detected by computing $(b_i - b_h)M_{ih}$ for each pair of bids, requiring $\frac{n^2-n}{2}$ comparisons. When equal bids have been detected, k extra rows might be inserted similar to the previous technique. As $n < k$ in most reasonable auction settings, the computational complexity per bidder remains $\mathcal{O}(nk)$ when bids are pairwise different. The exact complexity is $\mathcal{O}(nkT)$, where $T = n - |\{b_i\}_{i=1}^n| + 1$. This technique is generally less complex than the previous one (they are equally complex for the extreme case when all bids are equal). Due to the revelation of equal bids, there is no incentive for malicious bidders to use ties on purpose anymore. However, malicious bidders can try to "guess" bids, i.e., they submit various differing bids and hope for ties, because ties reveal opponents' bids.

Determining Ties (DET) Instead of trying to avoid ties, we can locate the position of ties. As mentioned before, ties only inhibit the protocol when they occur at the $(M+1)$st-highest bid. For this reason, "bad" ties always indicate the selling price. The following method marks ties if they prevent the regular protocol from working. $\sum_{i=1}^n \boldsymbol{b}_i - t\boldsymbol{e}$ is a vector that contains zeros if t bidders share the same bid at the corresponding position ($1 < t \leq n$). "Good" ties can be masked by adding $(n+1)\left(\mathsf{L}\sum_{i=1}^n \boldsymbol{b}_i - (t+u)\boldsymbol{e}\right)$ where $0 \leq u \leq M$ and $M+1 \leq t+u \leq n$. 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, the position of the tie (which is the selling price) has to be made invisible to losing bidders like in Section 5. This can be done by adding $(n^2+2n+1)(\mathsf{U}-\mathsf{I})\boldsymbol{b}_a$.

Concluding, this method requires the additional computation of indicators

$$\boldsymbol{v}'_{atu} = \left(\sum_{i=1}^n \boldsymbol{b}_i - t\boldsymbol{e} + (n+1)\left(\mathsf{L}\sum_{i=1}^n \boldsymbol{b}_i - (t+u)\boldsymbol{e}\right) + (n^2+2n+1)(\mathsf{U}-\mathsf{I})\boldsymbol{b}_a\right)\mathsf{R}^*_{atu} =$$
$$= \left((\mathsf{L}+(n+1)\mathsf{I})\sum_{i=1}^n \boldsymbol{b}_i - (nt+nu+2t+u)\,\boldsymbol{e} + (n^2+2n+1)(\mathsf{U}-\mathsf{I})\boldsymbol{b}_a\right)\mathsf{R}^*_{atu},$$

which increases the overall computational complexity to $\mathcal{O}(n^2kM)$. Information revelation is low compared with the previous two methods if we assume that ties happen "accidently" which can be justified by the fact that there is no gain by using equal bids strategically. Winning bidders learn that the selling

price was shared by t bidders and that there were u higher bids. In contrast to the previous two methods, not a single bid origin, i.e. a bidder's identity, is uncovered.

Suppose we have the following compilation of bids ($M = 1$, computation takes place in $\mathbb{Z}_{11}$):

$$\boldsymbol{b}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{b}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad .$$

The first two ($t = 2, u \in \{0, 1\}$) indicators look like this (before being masked for each bidder):

$$\begin{pmatrix} 0 \\ 2 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} + 5 \left(\begin{pmatrix} 0 \\ 2 \\ 2 \\ 4 \\ 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \right) = \begin{pmatrix} 10 \\ 0 \\ 9 \\ 10 \\ 8 \\ 8 \end{pmatrix} \xrightarrow{\times R^*_{1,2,0}} \begin{pmatrix} . \\ 0 \\ . \\ . \\ . \\ . \end{pmatrix}$$

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

For $t > 2$ the first difference contains no zeros, leading to random vectors.

5.3 Round vs. Computational Complexity

General fully private MPC is possible (in the computational model) when assuming weak robustness [35, 36]. This means that computational complexity could be drastically reduced by working with the currently most efficient scheme based on homomorphic encryption [37], which allows multiplication of encrypted values in three rounds. Different vectors that indicate prices by zeros can be multiplied into a single vector. This can be used to simplify the computation of indication vectors without any tie problems ($0 \leq u \leq M$).

$$\boldsymbol{v}_{au} = (\mathsf{L} - \mathsf{I}) \sum_{i=1}^{n} \boldsymbol{b}_i - u\boldsymbol{e}, \qquad \boldsymbol{v}_a = \left(\begin{pmatrix} \prod_{u=0}^{M} v_{au1} \\ \prod_{u=0}^{M} v_{au2} \\ \vdots \\ \prod_{u=0}^{M} v_{auj} \end{pmatrix} + \mathsf{U}\boldsymbol{b}_a \right) \mathsf{R}_a^*$$

This technique results in $\mathcal{O}(\log M)$ rounds after all. Additionally, the new structure of $\boldsymbol{v}_a$ enables binary search (in public-price mode, see Section 6) which furthermore decreases the computational complexity to $\mathcal{O}(\log k \log M))$. Please note, that bidders still need to submit k bid values.

However, MPC based on homomorphic encryption is currently only possible for factorization based encryption schemes like Paillier encryption [38]. In contrast to discrete logarithm based schemes, the joint generation of secret keys needed for such schemes is very inefficient [39–41], especially when requiring full privacy. There is (yet) no general MPC scheme based on ElGamal encryption.

6 Conclusion

We presented a novel cryptographic auction protocol where bidders jointly compute the auction outcome in a constant number of rounds (three when assuming the random oracle model). The price we pay for round complexity that does neither depend on the number of bidders n nor on the number of possible bids k is computational complexity that is linear in k. However, experimental results indicate that the computational amount and message sizes are manageable in many realistic settings, despite its linearity in k.

The protocol complies with the highest standard of privacy possible: it is safe for a single bidder no matter how many of the other participants collude. The only agent being able to discover who won the auction besides the concerned bidders is the seller. We are not aware of any auction protocol, that achieves a similar level of privacy. Only computationally unbounded adversaries can uncover information. When using verifiable secret sharing instead of homomorphic encryption (like in [32]), only bid statistics are revealed to less than $n - 1$ unbounded adversaries. As the protocol is publicly verifiable, malicious bidders that do not follow the protocol will be detected immediately and can be excluded from the set of bidders.

Table 1 shows the complexity of the protocol. When the selling price does not need to be protected ("public price"), the computational complexity can be reduced by just computing *one* value for all bidders that indicates the selling price p_w. Winning bidders can prove their claims to the seller by showing that $(\alpha_{iw}, \beta_{iw})$ is an encryption of Y. However, winning bidders are able to remain silent if they dislike the selling price (violating non-repudiation) in public-price mode. This could be circumvented by forcing all bidders to open their commitments for the selling price, thus proving to the seller whether they won or lost.

Price	Rounds	Computation (exponentiations)		
Private	$\mathcal{O}(1)$	INT: $\mathcal{O}(n^2k)$,	PRE: $\mathcal{O}(nkT)$,	DET: $\mathcal{O}(n^2kM)$
Public	$\mathcal{O}(1)$	INT: $\mathcal{O}(nk)$,	PRE: $\mathcal{O}(kT)$,	DET: $\mathcal{O}(nkM)$

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

Table 1. Protocol complexity (computation per bidder)

The protocol can be easily adapted to execute first-price or ascending (e.g. English) auctions. The latter might be useful in common-value scenarios where valuations interdepend. In the future, we intend to apply the presented techniques to solve tractable instances of combinatorial auctions like general multi-unit or linear-good auctions while maintaining full privacy.

References

1. Vickrey, W.: Counter speculation, auctions, and competitive sealed tenders. Journal of Finance **16** (1961) 8–37
2. Rothkopf, M.H., Teisberg, T.J., Kahn, E.P.: Why are Vickrey auctions rare? Journal of Political Economy **98** (1990) 94–109
3. Rothkopf, M.H., Harstad, R.M.: Two models of bid-taker cheating in Vickrey auctions. Journal of Business **68** (1995) 257–267
4. Sandholm, T.: Limitations of the Vickrey auction in computational multiagent systems. In: Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS), Menlo Park, CA, AAAI Press (1996) 299–306
5. Brandt, F., Weiß, G.: Vicious strategies for Vickrey auctions. In Müller, J., Andre, E., Sen, S., Frasson, C., eds.: Proceedings of the 5th International Conference on Autonomous Agents, ACM Press (2001) 71–72
6. Brandt, F., Weiß, G.: Antisocial agents and Vickrey auctions. In Meyer, J.J.C., Tambe, M., eds.: Intelligent Agents VIII. Volume 2333 of Lecture Notes in Artificial Intelligence (LNAI)., Springer (2001) 335–347 Revised papers from the 8th Workshop on Agent Theories, Architectures and Languages.
7. Brandt, F.: Cryptographic protocols for secure second-price auctions. In Klusch, M., Zambonelli, F., eds.: Cooperative Information Agents V. Volume 2182 of Lecture Notes in Artificial Intelligence (LNAI)., Springer (2001) 154–165
8. Wurman, P., Walsh, W., Wellman, M.: Flexible double auctions for electronic commerce: Theory and implementation. Decision Support Systems **24** (1998) 17–27
9. Franklin, M.K., Reiter, M.K.: The design and implementation of a secure auction service. IEEE Transactions on Software Engineering **22** (1996) 302–312
10. Abe, M., Suzuki, K.: M+1-st price auction using homomorphic encryption. In: Proceedings of the 5th International Conference on Public Key Cryptography (PKC). Volume 2274 of Lecture Notes in Computer Science (LNCS)., Springer (2002) 115–224

11. Abe, M., Suzuki, K.: Receipt-free sealed-bid auction. In: Proceedings of the 1st Information Security Conference (ISC). Volume 2433 of Lecture Notes in Computer Science (LNCS). (2002) 191–199
12. Baudron, O., Stern, J.: Non-interactive private auctions. In: Proceedings of the 5th Annual Conference on Financial Cryptography (FC). (2001) 300–313
13. Cachin, C.: Efficient private bidding and auctions with an oblivious third party. In: Proceedings of the 6th ACM Conference on Computer and Communications Security. (1999) 120–127
14. Harkavy, M., Tygar, J.D., Kikuchi, H.: Electronic auctions with private bids. In: Proceedings of the 3rd USENIX Workshop on Electronic Commerce. (1998) 61–74
15. Jakobsson, M., Juels, A.: Mix and match: Secure function evaluation via ciphertexts. In: Proceedings of the 6th Asiacrypt Conference. Volume 1976 of Lecture Notes in Computer Science (LNCS)., Springer (2000) 162–177
16. Kikuchi, H.: (M+1)st-price auction protocol. In: Proceedings of the 5th Annual Conference on Financial Cryptography (FC). Volume 2339 of Lecture Notes in Computer Science (LNCS)., Springer (2001) 351–363
17. Kikuchi, H., Harkavy, M., Tygar, J.D.: Multi-round anonymous auction protocols. In: Proceedings of the 1st IEEE Workshop on Dependable and Real-Time E-Commerce Systems. (1998) 62–69
18. Kikuchi, H., Hotta, S., Abe, K., Nakanishi, S.: Resolving winner and winning bid without revealing privacy of bids. In: Proceedings of the International Workshop on Next Generation Internet (NGITA). (2000) 307–312
19. Kudo, M.: Secure electronic sealed-bid auction protocol with public key cryptography. IEICE Transaction Fundamentals **E81-A** (1998)
20. Lipmaa, H., Asokan, N., Niemi, V.: Secure Vickrey auctions without threshold trust. In Blaze, M., ed.: Proceedings of the 6th Annual Conference on Financial Cryptography (FC). Volume 2357 of Lecture Notes in Computer Science (LNCS)., Springer (2002) to appear.
21. Naor, M., Pinkas, B., Sumner, R.: Privacy preserving auctions and mechanism design. In: Proceedings of the 1st ACM Conference on Electronic Commerce, ACM Press (1999) 129–139
22. Sako, K.: An auction protocol which hides bids of losers. In: Proceedings of the 3rd International Conference on Public Key Cryptography (PKC). Volume 1751 of Lecture Notes in Computer Science (LNCS)., Springer (2000) 422–432
23. Sakurai, K., Miyazaki, S.: A bulletin-board based digital auction scheme with bidding down strategy - Towards anonymous electronic bidding without anonymous channels nor trusted centers. In: Proceedings of the International Workshop on Cryptographic Techniques and E-Commerce. (1999) 180–187
24. Song, D.X., Millen, J.K.: Secure auctions in a publish/subscribe system. Available at http://www.csl.sri.com/users/millen/ (2000)
25. Viswanathan, K., Boyd, C., Dawson, E.: A three phased schema for sealed bid auction system design. In: Proceedings of the Australasian Conference for Information Security and Privacy (ACISP). Lecture Notes in Computer Science (LNCS) (2000) 412–426
26. Watanabe, Y., Imai, H.: Reducing the round complexity of a sealed-bid auction protocol with an off-line TTP. In: Proceedings of the 7th ACM Conference on Computer and Communications Security, ACM Press (2000) 80–86
27. Sakurai, K., Miyazaki, S.: An anonymous electronic bidding protocol based on a new convertible group signature scheme. In: Proceedings of the 5th Australasian Conference on Information Security and Privacy (ACISP2000). Volume 1841 of Lecture Notes in Computer Science (LNCS)., Springer (2000) 385–399

28. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC), ACM Press (1988) 1–10
29. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology **4** (1991) 161–174
30. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Advances in Cryptology - Proceedings of the 12th Annual International Cryptology Conference (CRYPTO). Volume 740 of Lecture Notes in Computer Science (LNCS)., Springer (1992) 3.1–3.6
31. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Advances in Cryptology - Proceedings of the 14th Annual International Cryptology Conference (CRYPTO). Volume 893 of Lecture Notes in Computer Science (LNCS)., Springer (1994) 174–187
32. Brandt, F.: A verifiable, bidder-resolved auction protocol. In Falcone, R., Barber, S., Korba, L., Singh, M., eds.: Proceedings of the 5th International Workshop on Deception, Fraud and Trust in Agent Societies (Special Track on Privacy and Protection with Multi-Agent Systems). (2002) 18–25
33. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31** (1985) 469–472
34. Tsiounis, Y., Yung, M.: On the security of ElGamal-based encryption. In: Proceedings of the 1st International Workshop on Practice and Theory in Public Key Cryptography (PKC). Volume 1431 of Lecture Notes in Computer Science (LNCS)., Springer (1998) 117–134
35. Brandt, F.: Social choice and preference protection - Towards fully private mechanism design. In: Proceedings of the 4th ACM Conference on Electronic Commerce, ACM Press (2003) 220–221
36. Brandt, F.: Private public choice. Technical Report FKI-247-03, Department for Computer Science, Technical University of Munich (2003) ISSN 0941-6358.
37. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Advances in Cryptology - Proceedings of the 18th Eurocrypt Conference. Volume 2045 of Lecture Notes in Computer Science (LNCS)., Springer (2001) 280–300
38. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology - Proceedings of the 16th Eurocrypt Conference. Volume 1592 of Lecture Notes in Computer Science (LNCS)., Springer (1999) 223–238
39. Algesheimer, J., Camenisch, J., Shoup, V.: Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In: Advances in Cryptology - Proceedings of the 22th Annual International Cryptology Conference (CRYPTO). Volume 2442 of Lecture Notes in Computer Science (LNCS)., Springer (2002) 417–432
40. Boneh, D., Franklin, M.: Efficient generation of shared RSA keys. In: Advances in Cryptology - Proceedings of the 17th Annual International Cryptology Conference (CRYPTO). Volume 1294., Springer (1997) 425–439
41. Damgård, I., Koprowski, M.: Practical threshold RSA signatures without a trusted dealer. In: Advances in Cryptology - Proceedings of the 18th Eurocrypt Conference. Volume 2045 of Lecture Notes in Computer Science (LNCS)., Springer (2001) 152–165