Secure and Private Auctions without Auctioneers

Felix Brandt
Institut für Informatik, Technische Universität München
80290 München, Germany, brandtf@in.tum.de

Technical Report FKI-245-02
February 2002

Abstract

Security and privacy have become crucial factors in auction design. Various schemes to ensure the safe conduction of sealed-bid auctions have been proposed recently. We introduce a new standard of security for auctions ("full privacy"), that prevents extraction of bid information despite any collusion of participants. This requirement is stronger than other common assumptions that prohibit the collusion of certain third-parties (e.g., distinct auctioneers). Full privacy is obtained by distributing shares of each bid on all bidders. The bidders then jointly compute the selling price without uncovering any additional information. Auctioneers are obsolete in these "bidder-resolved" auction protocols. The auction outcome cannot be changed by dishonest bidders; they can only inhibit the protocol. For this reason, we present a sub-protocol that detects dishonest bidders, so that they can be fined, which provides the incentive to follow the main protocol.

The major contribution of this work is the fully private Vickrey auction protocol that determines the second-highest bid without revealing any other information. As full privacy is our main goal, the drawback of our protocol is efficiency. For this reason, it is currently only applicable for high-security auctions with relatively few bidders in reasonable time.

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. Even the world's largest internet auction house `ebay` recently introduced a "private auction", in which bids are anonymous and only the seller and the winning bidder learn the result of the auction. Obviously, privacy in these auctions is very limited as it is up to the auction house whether the bids remain confidential. Additionally, non-public, high-revenue auctions like spectrum license auctions require a much higher level of protection.

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". In a regular first-price auction, there are few possibilities to cheat for the auctioneer if he has to announce the selling price at the end of the auction. He could declare a price greater than the highest bid, in

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

order to keep the good if he thinks the bids are not high enough. No bidder would be able to discover this form of deception. In a second-price or so-called Vickrey auction, things are worse. 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 in all sealed-bid auctions have to reveal their bids to the auctioneer. There are numerous ways to misuse these values by giving them away to other bidders or the seller [5, 4, 3]. It remains in the hands of the auctioneer whether the auction really is a *sealed*-bid auction.
Among the different auction protocols, the Vickrey auction [26] has received particular attention in recent times because it is "incentive-compatible", 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. This problem has been addressed several times in the literature [19, 18, 22] and it is now common knowledge 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 private valuations.
The protocol in this report 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.
The remainder of this report 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 introduces "bidder-resolved auctions". In Section 4, we propose a simple protocol that realizes a bidder-resolved 1st-price auction, which is followed by the enhanced, fully private protocol MB-SHARE. Finally, the Vickrey auction protocol YMB-SHARE, which complies with full privacy as well, is presented in Section 5. The report concludes with a brief overview of advantages and disadvantages of bidder-resolved auctions and an outlook in Section 6.

2 Related Work

There has been an extremely fast-growing interest in cryptographic protocols for auctions during the last years. In particular, Vickrey auctions, which are strategically equivalent to English auctions for bidders that privately evaluate a good, attracted much attention. Starting with the work by Franklin and Reiter [9], which introduced the basic problems, but disregarded the privacy of bids after the auction, many secure auction mechanisms have been proposed [1, 3, 6, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 24, 27, 28].
When taking away all the protocols that (in their current form) are not suitable for the secure execution of *second*-price auctions or reveal (partial) information after the auction is finished [9, 28, 21, 20, 11, 15, 27, 3], the remaining work can be divided into two categories.
Most of the publications rely on the (limited) security of distributed computation [12, 14, 13, 10, 24]. This technique requires m auctioneers, out of which a fraction (e.g., $\lfloor \frac{m-1}{3} \rfloor$) 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 sophisticated, but sometimes inefficient, techniques of secure multiparty function evaluation, mostly via distributed polyno-

mials. 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 general problem, because you can never rule out that all of them collude. This point of view is supported in a growing number of publications [16, 17, 25].
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 [1, 6] the third-party may not collude with any participant; in [16, 17] it is prohibited that the third-party and the auctioneer collude.
Concluding, all present work on secure Vickrey auctions more or less relies on the exclusion of third-party collusion, may it be auctioneers or other semi-trusted institutions. The technique we propose in this report is secure for a bidder, even if *all* other bidders collude.

3 General Assumptions

This section contains demands that our protocols will meet. Furthermore, we make several basic assumptions about bidders and collusions between them.

3.1 Privacy and Correctness

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 bidder and the seller learn the selling price and the seller finds out the winner's identity. As [21] pointed out, *anonymity* of the winner is crucial. Otherwise, a bidder that breaks a collusive agreement could be identified by his partners.
In several schemes, it is necessary that the auctioneer announces the selling price, in order to prevent the auctioneer from awarding the contract to a bogus bidder (violating *correctness*).

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, unlike some other protocols.

Correctness *The winner and the selling price are determined correctly.*

This requirement contains *non-repudiation* (the winning bidder cannot deny having made the winning bid). Bids are binding. Otherwise, bidders could control the selling price in first-price and second-price auctions by using subagents. Correctness also includes *robustness* (no set of malicious bidders can render the auction outcome invalid). If the auction protocol is interactive, this implies that missing bidder messages will not halt the auction process. In a weaker formulation of this property, needed for bidder-resolved auctions, bidders are able to falsify the auction result by not following the protocol, but each malicious bidder can be tracked down and fined or excluded from the set of bidders. When malicious bidders are detectable and fines are high enough, there should be no incentive to perturb the auction.

Of course, *efficiency* is also an important factor, but as we want to obtain full privacy, we regard efficiency as secondary. Privacy and correctness have to be ensured in a hostile environment, which is described by the following assumptions:

- Each agent (bidder or seller) can have arbitrarily many bidder sub-agents, controlled by him, in any auction.
- Up to $n-1$ bidders might share their knowledge and act as a team
- Any number of auctioneers or other third-parties might share their knowledge and give it away to bidders.

3.2 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 using a simple secret sharing scheme. Bidders divide their bids into n shares, keep one and send one share to each other bidder. Our protocols are designed in a way, that renders it impossible for a bidder to change the outcome of an auction or to gain knowledge by manipulating the shares, that have been entrusted to him. All he can do is nullify the auction, so that no winner will be determined. However, such a bidder is detectable and can be fined and/or excluded. For this reason, bidders cannot be anonymous. It has to be possible to make them responsible for their actions [2] .
The information sharing among bidders allows us to set a very high 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 do so, each of them abandons his own privacy, resulting in an open-cry auction.

Definition: A secure, bidder-resolved auction protocol complies with *full privacy* when no information on any bid can be retrieved unless all involved agents collude.

When using classical terms of secure multiparty computation [8], full privacy can be interpreted as $(n-1)$-privacy. A passive adversary that controls up to $n-1$ bidders is incapable of uncovering any information. Active adversaries, that mutilate the distributed computation will be detected by a sub-protocol, but only if they affected the outcome of the auction. This sub-protocol will reveal information about the highest bidder, which implies that fines have to be high enough to prevent deliberate disturbance. Alternatively, auctions could be repeated whenever malicious bidders disrupt the protocol. There would be no information revelation, but as we require an auction to be robust against active adversaries, we prefer the detection and fining of malicious bidders, even though it might reveal the winner and the selling price.
A threshold-scheme, that provides 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 adapt existing, successful schemes that were designed for m auctioneers like [10] or [12] because they rely on secure multiparty computation according to Ben-Or, Goldwasser and Widgerson [2], which in turn provides at most insufficient $\lfloor \frac{n}{2} \rfloor$-privacy due to the multiplication of degree n polynomials.
Like in most recent protocols, we define an ordered set of k possible prices (or valuations) $\{p_1, p_2, \ldots, p_k\}$. In contrast to ascending auctions, bids have an upper

[2] We currently review the utilization of prior shared deposits.

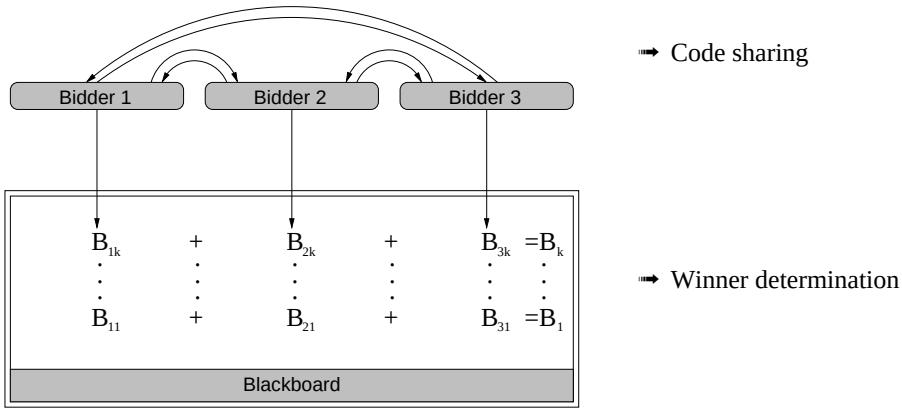

Figure 1: B-SHARE

limit. Each bidder submits k binary bids denoting whether he is willing to pay a given price or not. All proposed protocols have in common that bidders jointly compute a commutative *bid function* $f(X_1, X_2, \ldots, X_n)$ for each price p_j.
Bidders communicate with each other through private and public channels, i.e., messages are encrypted and signed.

4 1st-price Auctions

In a first-price sealed-bid auction, each bidder submits a sealed bid and the highest bidder wins the auction. The price he has to pay is the amount of his bid. Thus, n bidders need to secretly compute the maximum of n values.

4.1 Interactive Protocol (Dutch auction)

There already is a perfectly secure and private, interactive first-price auction protocol: the Dutch auction. The auctioneer announces a decreasing bid from round to round starting with the highest possible price. The first bidder that stops the auction by expressing his willingness to pay is awarded the contract for the amount of the actual bid. This might take some time (depending on k), but no information except the selling price is revealed.

4.2 Protocol B-SHARE

Serving as a simple example of bidder-resolved auctions, this protocol will not meet all demanded criteria specified in Section 3. B-SHARE's bid function $f_1 : G^n \mapsto G$ is defined on the finite, Abelian group $\langle G, + \rangle$.

$$f_1(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} X_i \tag{1}$$

It is jointly computed by using additive shares of each bid code. The ith additive share of x is denoted by x^{+i}.
Harkavy et al proposed this kind of protocol in [10]. However, they distributed the bids on m auctioneers.

		Rounds	Messages	Bandwidth
B-SHARE	overhead	-	-	-
	main	$O(1)$	$O(n)$	$O(nk)$
MB-SHARE	overhead	-	-	-
	main	$O(n)$	$O(n)$	$O(nk)$
YMB-SHARE	overhead	$O(n)$	$O(n)$	$O(n^2k)$
	main	$O(n)$	$O(n)$	$O(nk)$

Table 1: Protocol complexity (messages and bandwidth per bidder)

4.2.1 Protocol Sequence

The following protocol steps have to be executed by an arbitrary bidder a. $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$. All calculations are done in the finite Abelian group $\langle G, + \rangle$. 0 is the neutral element of $\langle G, + \rangle$.

- **Create codes**
  - Choose Y_{aj} for each j and commit to Y_{aj} by sending a cryptographic hash to the seller.
- **Share codes**
  - Choose b_{aj}^{+i} for each j and i, so that

$$\sum_{i=1}^{n} b_{aj}^{+i} = \begin{cases} Y_{aj} & \text{if bidder } a \text{ is willing to pay } p_j \\ 0 & \text{else} \end{cases} .$$

  - Send b_{aj}^{+i} for each j to bidder i for each $i \neq a$.
  - Receive b_{ij}^{+a} for each $i \neq a$ and j.
  - Publish $b_j^{+a} = \sum_{i=1}^{n} b_{ij}^{+a}$ for each j.
  - Compute $B_j = \sum_{a=1}^{n} b_j^{+a}$ for each j by using the published b_j^{+a}.
- **Winner determination**
  - If $B_j = Y_{aj}$ for any j, then bidder a won the auction. The selling price $p_{\min\{j \mid B_j = 0\} - 1}$ is visible to all bidders. Only the seller and the winning bidder can learn the winner's identity.

4.2.2 Analysis

The intermediate sums need to be published simultaneously. This can be achieved by using a bit commitment technique, e.g., via a cryptographic hash function. As a consequence, bidders can only nullify auction by not following the protocol. Well directed manipulation is impossible.
Table 1 shows the number and lengths of messages that have to be sent by a single bidder. Please note that the entire protocol is finished after two rounds (sending the shares and publishing the sums). Fixed execution time is the major advantage over the interactive Dutch auction protocol.

Convenient choices for the finite group $\langle G, + \rangle$ are $\langle Z_{2^l}, + \rangle$ or $\langle \{0,1\}^l, \text{XOR} \rangle$. There is a very small probability of failure depending on n and l if two or more bidders chose the same code for the same price or if several codes add up to zero by chance, but this is negligible for large l.
Any bidder that does not follow the prescribed protocol *and* changes its outcome has to be detectable, so that he can be fined or punished. When applying B-SHARE, there are two ways for a bidder to disrupt the auction.

- *Idleness*: A bidder does not send the required messages in time. Idle bidders can be punished and/or excluded from the set of bidders.
- *False Computation*: A bidder does not carry out the prescribed computation correctly. If the outcome is affected, the bidder can be tracked down.

The so-called "Fault Detection Protocol", has to be executed whenever the seller claims that he cannot identify a winner. In this case, all signed messages, i.e., all b_{aj}^{+i}, have to be published, beginning at the highest j. This method yields either

1. a lying seller,
2. a bidder that committed *False Computation*, or
3. a tie between two or more bidders.

As soon as one of these cases is detected, the protocol is halted. In the first two cases or if a bidder does not provide the messages required by the Fault Detection Protocol, the corresponding agent has to be fined and/or excluded from a possible re-auctioning of the same item. The Fault Detection Protocol only reveals the identity of the highest bidder; losing bids will not be opened. For this reason, the fines for malicious bidders should be high enough to prevent bidders from "buying" information on the identity of the highest bid.
As a consequence of the Fault Detection Protocol, a tie between two or more bidders results in the revelation of the bidders' identities and their (identical) bids. However, this case demands a random-based decision to determine the winner, which requires the participation of all bidders. One possible solution is to enlarge the bidding set to nk values, so that bidder i is only allowed to bid a multiple of i. The bidder ordering has to be arranged at random beforehand, which enables fair winner determination.
Besides "only" providing $(n-2)$-privacy, the protocol has another major flaw. The second-highest bid can be read by the winner of the auction [14]. More generally, the cth highest bid can be read by a collusion of the $c-1$ highest bidders. We will fix that in the subsequent protocol.

4.3 Protocol MB-SHARE

In order to mask the sums of the previous protocol, we multiply them with a shared random multiplier $M_j = \prod_{i=1}^{n} m_j^{\times i}$ that is not known to any of the bidders. A similar solution was proposed in [14]. However, the implementation provides at most $\lfloor \frac{n}{2} \rfloor$-privacy due to the reasons specified in Section 3.2. Our protocol uses a one-way function and the propagation of values from bidder to bidder (*ring transfer*) to realize a multiplication.
The following function $f_2 : F^n \mapsto F$ is used as the bid function (g is a generator in the multiplicative group of the finite field F, i.e., $F = \{g^i \mid i \in \mathbb{Z}\}$).

$$f_2(X_1, X_2, \ldots, X_n) = g^{\sum_{i=1}^{n} X_i M} \tag{2}$$

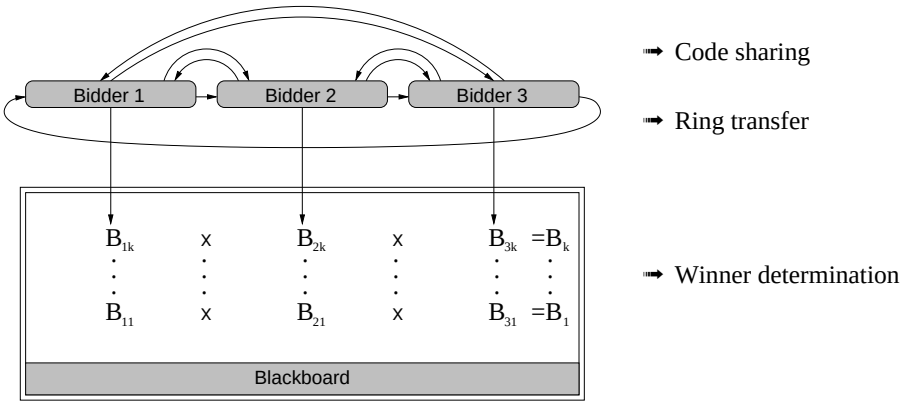

Figure 2: MB-SHARE

All bidders spread additive shares of their X_i. Then, each bidder calculates $g^{x_j^{+a}}$ with x_j^{+a} being the sum of bidder a's additive shares. This value is handed from bidder to bidder (ring transfer) and raised to the power of $m_j^{\times a}$ (the multiplier share). The last bidder publishes the resulting value $g^{x_j^{+a}\prod_{i=1}^{n} m_j^{\times i}}$. When multiplying all published values, this yields

$$\prod_{a=1}^{n} g^{x_j^{+a}\prod_{i=1}^{n} m_j^{\times i}} = g^{\sum_{a=1}^{n}(x_j^{+a}\prod_{i=1}^{n} m_j^{\times i})} = f_2(X_1, X_2, \ldots, X_n) \; .$$

The exponential (one-way) function ensures privacy of the shared multipliers, based on the intractability of the discrete logarithm problem. Function f_2 was originally developed for the secure Vickrey auction protocol YMB-SHARE and has further advantageous features that will be explained in Section 5.2.
In order to enable ring transfer, we need an ordering on bidders. $s(i)$ returns the successor to bidder i.

$$s(i) = \begin{cases} i+1 & \text{if } i < n \\ 1 & \text{else} \end{cases}$$

4.3.1 Protocol Sequence

The following is the protocol for an arbitrary bidder a. $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$. All calculations are done in the finite field F, e.g., GF(2^l). $b_a \in \{1, 2, \ldots, k\}$ denotes bidder a's bid.

- **Create codes/ Commit to bid**
  - Choose Y_{aj} and $m_j^{\times a} \neq 0$ for each j and a random number r_a.
  - Commit to bid b_a by sending cryptographic hashes of $b_a + r_a$ and r_a to the seller.
- **Share codes**
  - Choose b_{aj}^{+i} for each j and i, so that $\sum_{i=1}^{n} b_{aj}^{+i} = \begin{cases} Y_{aj} & \text{if } b_a \geq j \\ 0 & \text{else} \end{cases}$.
  - Send b_{aj}^{+i} for each j to bidder i for each $i \neq a$.

- Receive b_{ij}^{+a} for each $i \neq a$ and j.

- **Ring transfer**
  - Compute ${}_{n-1}B_{aj} = g^{\sum_{i=1}^{n} b_{ij}^{+a} m_j^{\times a}}$ for each j and send them to bidder $s(a)$.
  - When receiving ${}_rB_{ij}$, compute ${}_{r-1}B_{ij} = ({}_rB_{ij})^{m_j^{\times a}}$. If $r > 1$, send it to bidder $s(a)$; else, publish $B_{ij} = {}_0B_{ij}$.
  - Compute $B_j = \prod_{i=1}^{n} B_{ij}$ for each j by using the published B_{ij}.

- **Winner determination**
  - The selling price $p_{\min\{j|\, B_j=1\}-1}$ is visible to all bidders. The winning bidder authenticates to the seller by secretly sending r_a.

4.3.2 Analysis

Ring transfer requires n additional rounds, but after all the message complexity is unchanged (Table 1). Due to the intractability of the discrete logarithm problem and the masking multiplications, the protocol is now fully private.

Theorem: MB-SHARE is fully private (except the declaration of the highest bid).

Proof: As we can subsume a collusion of $n-1$ bidders to one single bidder, it suffices to show, that in an auction with two bidders, bidder 2 cannot reveal bidder 1's bid b_1. We assume that $b_1 < b_2$ because only losing bids are protected in MB-SHARE. Bidder 2 needs to test whether $b_{1j}^{+1} = -b_{1j}^{+2}$ for each j. There are two possibilities to achieve this. First, bidder 2 can try to extract b_{1j}^{+1} by using any combination of values known to him: b_{2j}^{+1}, b_{2j}^{+2}, b_{1j}^{+2}, m_{2j}, $B_j = g^{(b_{1j}^{+1}+b_{1j}^{+2}+b_{2j}^{+1}+b_{2j}^{+2})m_{1j}m_{2j}}$, and $g^{(b_{1j}^{+1}+b_{2j}^{+1})m_{1j}}$. This is not feasible unless the discrete logarithm problem can be solved. Secondly, he can try to compute $g^{(-b_{1j}^{+2}+b_{1j}^{+2}+b_{2j}^{+1}+b_{2j}^{+2})m_{1j}m_{2j}}$ and compare it with B_j. However, this is impossible because m_{1j} is unknown to bidder 2 and cannot be extracted from the values known to him. □

Besides "Idleness" and "False Computation", mentioned in Section 4.2.2, there is another possibility to disrupt the auction in this protocol.

- *Concealment of Victory*: A bidder remains silent even though he won the auction. Such a bidder will be identified by the Fault Detection Protocol.

The Fault Detection Protocol works like in the B-SHARE protocol with the only difference that intermediate ring transfer values ${}_rB_{aj}$ have to be published as well.

5 2nd-price Auctions

The 2nd-price sealed-bid auction works like the 1st-price auction with the only difference that the winning bidder pays the amount of the second-highest bid. In an optimal implementation, only the winning bidder (and the seller) can read the second-highest bid.

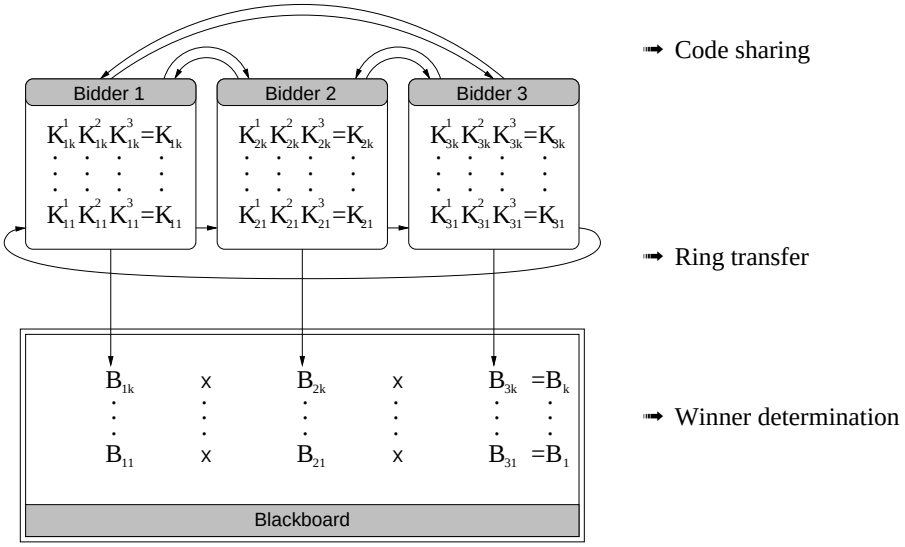

Figure 3: YMB-SHARE

5.1 Interactive Protocol ASCENDING MB-SHARE

It is possible to perform the previous protocol in an interactive fashion in order to hide the highest bid. This enables the execution of interactive Vickrey auctions. In this protocol, bidders share their bids iteratively for each price p_j beginning at the lowest price p_1. The execution of the protocol is stopped when a bidder claims (and proves) that he won the auction. This reveals only the selling price, but not how far the winner would have gone. When valuations depend on other bidders' valuations (*correlated value* model), this will lead to a higher revenue like in English auctions. However, this protocol is very slow as it takes $O(nk)$ messages to determine the winner. Kikuchi et al mentioned this type of protocol in [13] (but shared the information among distinct auctioneers).
Other interactive protocols that determine a pre-committed second-highest bid are presented in [3]. They do not necessarily need an auctioneer, but reveal partial information.

5.2 Protocol YMB-SHARE

In this protocol, each bidder has two different codes Y ("yes") and N ("no") for each price, denoting whether he is willing to pay at the given price or not. Bidders submit shares of their bids B_{aj} that are either Y_{aj} or N_{aj} and jointly compute $B_j = f_2(B_{1j}, B_{2j}, \ldots, B_{nj})$ for each price j. Personalized keys $K_{ij} = f_2(N_{1j}, N_{2j}, \ldots, Y_{ij}, \ldots, N_{nj})$ are jointly computed for each bidder i and price j, so that in the end only bidder i knows the value of K_{ij}. By comparing his keys with the published B_j, a bidder can find out whether he won the auction.
In order to prevent manipulation of keys, the yes-value for each bidder is jointly created by all bidders and the no-value $N_{aj} = -Y_{aj}$ can be derived by inverting all shares.

5.2.1 Protocol Sequence

Like in the previous sections, this is the step-by-step protocol specification for bidder a. $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$. All calculations are done in the finite field F, e.g., GF(2^l). g is a generator in the multiplicative group of F.

- **Create shared codes**
  - Choose $y_{ij}^{+a} \neq 0$ for each i and j and $m_j^{\times a}$ for each j.
- **Compute keys (using ring transfer)**
  - Compute ${}_nK_{ij}^{\times a} = g^{(y_{ij}^{+a} - \sum_{h \neq i} y_{hj}^{+a}) m_j^{\times a}}$ for each j and i and send them to bidder $s(a)$.
  - When receiving ${}_rK_{ij}^{\times h}$:
    If $r = 0$, set $K_{aj}^{\times h} = {}_0K_{ij}^{\times h}$ and commit to $K_{aj}^{\times h}$ by sending a cryptographic hash to the seller.
    Else, compute ${}_{r-1}K_{ij}^{\times h} = ({}_rK_{ij}^{\times h})^{m_j^{\times a}}$ and send it to bidder $s(a)$ if $r > 2$ or to bidder i if $r = 1$.
  - Compute $K_{aj} = \prod_{i=1}^{n} K_{aj}^{\times i}$ for each j.
- **De-share codes/ Share Bids**
  - Send y_{ij}^{+a} for each j to bidder i for each $i \neq a$.
  - Compute $Y_{aj} = \sum_{i=1}^{n} y_{aj}^{+i}$ and $N_{aj} = -Y_{aj}$ for each j.
  - Choose b_{aj}^{+i} for each j and i, so that $\sum_{i=1}^{n} b_{aj}^{+i} = \begin{cases} Y_{aj} & \text{if } b_a \geq j \\ N_{aj} & \text{else} \end{cases}$.
  - Send b_{aj}^{+i} for each j to bidder i for each $i \neq a$.
  - Receive b_{ij}^{+a} for each $i \neq a$ and j.
- **Ring transfer**
  - Compute ${}_{n-1}B_{aj} = g^{\sum_{i=1}^{n} b_{ij}^{+a} m_j^{\times a}}$ for each j and send them to bidder $s(a)$.
  - When receiving ${}_rB_{ij}$, compute ${}_{r-1}B_{ij} = ({}_rB_{ij})^{m_j^{\times a}}$. If $r > 1$, send it to bidder $s(a)$; else, publish $B_{ij} = {}_0B_{ij}$.
  - Compute $B_j = \prod_{i=1}^{n} B_{ij}$ for each j by using the published B_{ij}.
- **Winner determination**
  - If $B_j = K_{aj}$ for any j, then bidder a won the auction. He then contacts the seller and authenticates by supplying the signed messages containing $K_{aw}^{\times i}$ for each i and $w = \min\{j | \, B_j = K_{aj}\}$. The selling price is p_{w-1}.

5.2.2 Analysis

The combination of joint code creation, intractability of the discrete logarithm, and immutability of f_2, provide full privacy in the second-price auction protocol YMB-SHARE.

Theorem: YMB-SHARE is fully private.

Proof: Again, we show that in an auction with two bidders, bidder 2 cannot reveal bidder 1's bid b_1. There are three ways to reveal b_1. First of all, testing if $b_{1j}^{+1} + b_{1j}^{+2} = y_{1j}^{+1} + y_{1j}^{+2}$ cannot be realized, because b_{1j}^{+1} and y_{1j}^{+1} are unknown values and cannot be extracted from the one-way function f_2. Secondly, bidder 2 can try to derive $f_2(N_{1j}, N_{2j})$ from $K_{2j} = f_2(N_{1j}, Y_{2j})$ or $f_2(B_{1j}, Y_{2j})$ from $B_{2j} = f_2(B_{1j}, B_{2j})$, which both would uncover b_1 if $b_1 > b_2$. This is prevented by the masking multiplier m_{1j}. We say function f_2 is *immutable*. Finally, bidder 2 could try to compute $f_2(N_{1j}, N_{2j})$ *instead* of K_{2j}. He cannot win the auction this way, but as we assume that each bidder can have sub-agents, he could assign this job to a sub-agent. However, this is impossible, because Y_{1j} is jointly created by all bidders and $N_{1j} \neq Y_{1j}$ (with a very high probability). □

It would be possible to use the same multiplier M for all j. However, this would simplify the calculation of the discrete logarithm and thus make it easier to uncover M. Please note that if the discrete logarithm problem is solved (for the right j), only the amount of the highest bid can be read. All other bids enjoy unconditional $(n-2)$-privacy.
In contrast to most other protocols, including the previous two, the selling price is only visible to the winning bidder and the seller.
Table 1 shows the message and round complexity of the protocol. The computation of personalized keys for each bidder does not increase the number of messages, but results in a high demand for bandwidth ($O(n^2k)$). On the other hand, the same keys can be used for umpteen auctions with the same set of bidders. This requires commitment to an additional multiplier by each bidder at the beginning of an auction to prevent the seller from using the personalized keys. If the keys are used only once, the huge amount of numbers to exponentiate can be drastically reduced by substituting ${}_{r-1}K_{ij}^{\times h}$ with an arbitrary random number when $j \leq b_a$.
To give an example, we will compute the bandwidth demand of a typical high-security auction. Let us consider an auction with ten bidders ($n = 10$) and 200 possible prices ($k = 200$)[3]. We use 1024-bit numbers to ensure that f_2 is indeed a one-way function ($l = 1024$). Each seller has to send the following amount of data.

- overhead: $n^2kl + (n-1)kl$ bits $=$ 2.79 Mbytes
- main: $(n-1)kl + nkl$ bits $=$ 486 Kbytes

Like in the previous two protocols, ties result in an activation of the Fault Detection Protocol, which reveals the highest and second-highest bid (which are equal in this case) and their origins. Possible solutions are the partitioning of the bid set as described in Section 4.2.2 or the omission of the Fault Detection Protocol and re-start of the auction. As a matter of fact, there are protocols regarding an auction with equal winning bids as an auction with no winner at all.
The approach we chose to detect the second-highest bid cannot be transfered to uniform price auctions (sometimes called $(M+1)$st-price auctions) without major changes.

[3]Usually the number of different prices or valuations is much lower than one would expect, e.g., Lipmaa et al argue that $k \leq 500$ is sufficient for most auctions [16].

6 Conclusion

We presented a novel kind of secure and private auction protocols, where information is shared among bidders. The protocols comply with the highest standard of privacy possible: they are safe for a single bidder no matter how many of the participants collude. Malicious bidders, that nullify the auction outcome, can always be detected and fined.
Besides the fully private first-price auction protocol MB-SHARE, the main contribution of this report is the secure Vickrey auction protocol YMB-SHARE, in which bidders jointly compute personal keys for each bidder. Applying these keys, only a bidder can discover whether he won the auction or not. We are not aware of any Vickrey auction protocol, that achieves a similar level of privacy.
The drawback of YMB-SHARE is efficiency. Currently, it can take hours, if not days, to decide auctions with a high number of bidders. The execution time could be greatly reduced, if we did not have to rely on the discrete logarithm problem when computing f_2. Furthermore, the intractability of the discrete logarithm is the only cryptographic assumption made. As a consequence, the main part of our future research goes into the development of different functions f, possibly by using non-associative operations, which could result in a more efficient, unconditional protocol. Another option to optimize the protocol might be to use binary radix representations of bids, in order to reduce the bandwidth complexity to $O(n^2 \log k)$.

References

[1] O. Baudron and J. Stern. Non-interactive private auctions. In *Pre-Proceedings of the 5th Annual Conference on Financial Cryptography*, pages 300–313, 2001.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC-88)*, pages 1–10, 1988.

[3] F. Brandt. Cryptographic protocols for secure second-price auctions. In M. Klusch and F. Zambonelli, editors, *Cooperative Information Agents V*, volume 2182 of *Lecture Notes in Artificial Intelligence*, pages 154–165, Berlin et al., 2001. Springer.

[4] F. Brandt and G. Weiß. Antisocial agents and Vickrey auctions. In *Proceedings of the 8th workshop on Agent Theories, Architectures and Languages*, 2001.

[5] F. Brandt and G. Weiß. Vicious strategies for Vickrey auctions. In *Proceedings of the 5th International Conference on Autonomus Agents*, pages 71–72. ACM Press, 2001.

[6] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 120–127, 1999.

[7] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, pages 65–75, 1988.

[8] M. Franklin, Z. Galil, and M. Yung. An overview of secure distributed computing. Technical Report TR CUCS-008-92, Columbia University, 1992.

[9] M. Franklin and M. Reiter. The design and implementation of a secure auction service. *IEEE Trans. on Software Engineering*, 22(5):302–312, 1996.

[10] M. Harkavy, J. Tygar, and H. Kikuchi. Electronic auctions with private bids. In *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pages 61–74, 1998.

[11] M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts. In *Proceedings of Asiacrypt-00*, pages 162–177, 2000.

[12] H. Kikuchi. (M+1)st-price auction protocol. In *Proceedings of Financial Cryptography (FC 2001)*, 2001.

[13] H. Kikuchi, M. Harkavy, and J. Tygar. Multi-round anonymous auction protocols. In *Proceedings of the First IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, pages 62–69, 1998.

[14] H. Kikuchi, S. Hotta, K. Abe, and S. Nakanishi. Resolving winner and winning bid without revealing privacy of bids. In *Proceedings of the International Workshop on Next Generation Internet (NGITA)*, pages 307–312, 2000.

[15] M. Kudo. Secure electronic sealed-bid auction protocol with public key cryptography. *IEICE Trans. Fundamentals*, E81-A(1), 1998.

[16] H. Lipmaa, N. Asokan, and V. Niemi. Secure Vickrey auctions without threshold trust. In *Proceedings of the 6th Annual Conference on Financial Cryptography*, 2002. to appear.

[17] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and meachnism design. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, 1999.

[18] M. Rothkopf and R. Harstad. Two models of bid-taker cheating in Vickrey auctions. *Journal of Business*, 68(2):257–267, 1995.

[19] M. Rothkopf, T. Teisberg, and E. Kahn. Why are Vickrey auctions rare? *Journal of Political Economy*, 98(1):94–109, 1990.

[20] K. Sakurai and S. Miyazaki. A bulletin-board based digital auction scheme with bidding down strategy - towards anonymous electronic bidding without anonymous channels nor trusted centers. In *Proc. International Workshop on Cryptographic Techniques and E-Commerce*, pages 180–187, 1999.

[21] K. Sakurai and S. Miyazaki. An anonymous electronic bidding protocol based on a new convertible group signature scheme,. In *Proc. ACISP2000. Fifth Australasian Conference on Information Security and Privacy*, Lecture Notes in Computer Science, 2000.

[22] T. Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In *Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS-96)*, pages 299–306, Menlo Park, CA, 1996. AAAI Press.

[23] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[24] D. Song and J. Millen. Secure auctions in a publish/subscribe system. Available at http://www.csl.sri.com/users/millen/, 2000.

[25] S. Stubblebine and P. Syverson. Fair on-line auctions without special trusted parties. In *Proceedings of Financial Cryptography (FC 1999)*, volume 1648 of *Lecture Notes in Computer Science*, pages 230–240, 1999.

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

[27] K. Viswanathan, C. Boyd, and E. Dawson. A three phased schema for sealed bid auction system design. In *Australasian Conference for Information Security and Privacy (ACISP 2000)*, Lecture Notes in Computer Science, pages 412–426, 2000.

[28] Y. Watanabe and H. Imai. 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*, pages 80–86. ACM Press, 2000.

[29] A. Yao. How to generate and exchange secrets. In *Proceedings of the 27th Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society Press, 1986.