Cryptographic Protocols for Secure Second-Price Auctions

Felix Brandt

Institut für Informatik, Technische Universität München
80290 München, Germany
Tel. +49-89-28928416
brandtf@in.tum.de

Abstract. In recent years auctions have become more and more important in the field of multiagent systems as useful mechanisms for resource allocation, task assignment and last but not least electronic commerce. In many cases the Vickrey (second-price sealed-bid) auction is used as a protocol that prescribes how the individual agents have to interact in order to come to an agreement. The main reasons for choosing the Vickrey auction are the existence of a dominant strategy equilibrium, the low bandwidth and time consumption due to just one round of bidding and the (theoretical) privacy of bids. This paper specifies properties that are needed to ensure the accurate and secret execution of Vickrey auctions and provides a classification of different forms of collusion. We approach the two major security concerns of the Vickrey auction: the vulnerability to a lying auctioneer and the reluctance of bidders to reveal their private valuations. We then propose a novel technique that allows to securely perform second-price auctions. This is achieved using the announcement of encrypted binary bidding lists on a blackboard. Top-down, bottom-up and binary search techniques are used to interactively find the second highest bid step by step without revealing unnecessary information.

1 Introduction

The area of multiagent systems (e.g., [6, 16, 24]), which is concerned with systems composed of technical entities called agents that interact and in some sense can be said to be intelligent and autonomous, has achieved steadily growing interest in the last years. Electronic commerce, automated resource allocation and task assignment are topics that are of major interest in the agent community. As auctions are very flexible and efficient tools that can be used to address these problems, it has become common practice to apply well known results and insights from auction theory (e.g., [14, 15]) and well understood auction protocols like the English auction, the Dutch auction, and the Vickrey auction. Among the different protocols, the Vickrey auction [22] (also known as second-price sealed-bid auction) has received particular attention within the multiagent community (e.g., [9, 5, 23, 4, 3, 2]). due to three main characteristics:

- it requires low bandwidth and time consumption
- it possesses a dominant strategy, namely, to bid one's true valuation[1]
- it is a sealed-bid auction; bids (expressing private values) remain secret

These characteristics make the Vickrey auction protocol particularly appealing from the point of view of automation. The Vickrey auction, in its original formulation and as it is used for *selling goods* or *resource allocation*, works as follows: each bidder makes a sealed bid expressing the amount he is willing to pay, and the bidder who submits the highest bid wins the auction; the price to be payed by the winner is equal to the second highest bid. In *task assignment scenarios* the Vickrey auction works exactly the other way round (and for that reason is often referred to as reverse Vickrey auction): each bidder willing to execute a task makes a bid expressing the amount he wants to be payed for task execution, and the bidder submitting the lowest bid wins the auction; the winner receives an amount equaling the second lowest bid (and his payoff thus is the second lowest bid minus his prime costs for execution). If there is more than one winning bid, the winner is picked randomly. As the standard and the reverse auction are fully analogical, all presented considerations and techniques do hold for both formulations of the Vickrey auction.

Despite its impressive theoretical properties, the Vickrey auction is rarely used in practice. This problem has been addressed several times [18, 19, 17] and it is an undisputed fact that the Vickrey auction's sparseness is due to two major reasons. First, the proper execution of a Vickrey auction depends on the truthfulness of the auctioneer. The highest bidder has to trust the auctioneer when he is told the second highest bid. There is a great risk of an insincere auctioneer overstating the second highest bid on purpose to gain more money (either for himself or his client, the seller). Secondly, bidders are usually reluctant to reveal their true private values as is required by the dominant strategy. In many scenarios auctions are not isolated events, but rather parts of a whole series of negotiations. Valuations of goods or tasks are sensitive information that agents intend to keep private. Even if the transfer of the sealed bids is fully protected applying encryption techniques, it remains unknown how the auctioneer treats this confidential information. The privacy of sealed bids is also a substantial topic in regular first-price auctions, but it is of even more importance in second-price auctions since bids represent unaltered private values. This paper addresses both crucial weaknesses of the Vickrey auction. We present a technique that ensures the inability of the auctioneer to manipulate the outcome of the auction and we develop methods that reduce the auctioneer's knowledge of bids to a minimum in order to weaken his position in collusive agreements.

The paper is structured as follows. Section 2 summarizes existing efforts in the field of secure Vickrey auctions and explains why we restrict ourselves to a single auctioneer. Section 3 defines vital attributes that ensure a smooth auction conduction and specifies significant types of collusions. The foundations for the new secure auction protocol are set up in Section 4 and Section 5 introduces and analyzes three methods that hide unneeded information from the auctioneer. Finally, Section 6 concludes the paper with a brief overview of advantages and disadvantages of the proposed technique.

[1] if bidders are risk-neutral and have private valuations of goods or tasks.

2 Related Work

Franklin and Reiter were among the first to address electronic auction security[7]. They covered many basic problems, combined cryptographic primitives such as secret sharing, digital cash and multicasts, and introduced their own primitive called “verifiable signature sharing”.

There are only very few publications devoted to second-price auctions [8, 12, 11] and all of them (as most of the first-price auction papers [21, 1, 13, 10]) rely on the (limited) security of distributed computation. This technique requires m auctioneers, out of which $\lfloor \frac{m-1}{3} \rfloor$ 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 polynomials. However, a collusion of e.g., three out of five auctioneer servers can already exploit the bidders' trust. It would therefore be desirable to remove the trust in the auctioneer(s) entirely which is the main goal of this paper.

3 Properties of a secure Vickrey auction service

We consider a situation where one seller and n bidders or buyers intend to come to an agreement on the selling of a good or task. The auctioneer is an agent that acts as an intermediary between the seller and the bidders and fixes the selling price. In the following, we will set up properties that are required for flawless conductions of second-price sealed-bid auctions. These properties are divided into two categories. The first one deals with the accurate execution of the auction (Figure 1) whereas the second one defines rules that guarantee the privacy of confidential information (Figure 2).

E1 The auctioneer is not capable of determining a false winner.
E2 The auctioneer is not capable of manipulating the selling price upwards.
E3 The auctioneer is not capable of manipulating the selling price downwards.
E4 The highest bidder cannot deny having made the highest bid.
E5 The second highest bidder cannot deny or alter his bid.
E6 The auction process cannot be paralyzed by malicious bidders.

Fig. 1. Auction properties (execution)

Figure 1 and 2 specify the properties of an ideal Vickrey auction. There is yet no protocol that meets all these demands and it seems arguable whether the second highest bid can actually be determined without revealing any information at all. Information that has to be revealed by the auctioneer should be kept to a minimum to prevent him from providing unauthorized persons with these sensitive data.

We will not consider the possibility of a malicious auctioneer, that is an auctioneer that deliberately hampers the auction mechanism. Under the assumption that **E1**-**E3** are

P1 the bids and the corresponding bidders' identities are unknown prior to the opening of the bids.
P2 the bids and the corresponding bidders' identities remain unknown even after the auction process is finished (except the declaration of the second highest bid).

Fig. 2. Auction properties (privacy of information)

fulfilled, this behavior cannot result in an incorrect auction outcome, but only in no result at all. As a consequence, bidders could choose another auctioneer and re-auction the item. Furthermore, we will not approach the problem how to enforce the payment from the winning bidder. See [7] for a discussion of this issue involving electronic money. Besides, we assume that all bids are digitally signed and no bidder can act on the behalf of another.

An interesting question that arises when applying Vickrey auctions is which information has to be declared by the auctioneer after the bid submission and evaluation period is finished: the winner's identity, the selling price or both of the above?
It is inevitable to declare the winning price in a secure Vickrey auction protocol in order to prevent the auctioneer from awarding the contract to a bogus bidder. Each bidder can compare the declared price with his bid and complain if his bid is higher than the winning price and he has not been notified by the auctioneer. The bidder's identity, however, can normally be kept as a secret to the winner and the auctioneer. Yet, this requires that the declared selling price is undoubtedly the actual second highest bid.

To enable analysis of our auction protocol, we will consider every reasonable form of collusive agreements. We distinguish the following types of collusion:

- auctioneer/seller (A/S)
- auctioneer/bidder(s) (A/B)
- bidder/bidder (B/B)

B/B collusion can be seen as the most common type of collusion. As the English auction, the Vickrey auction is in particular vulnerable to B/B collusions, i.e., agents that team up to eliminate rivalry, resulting in lower selling prices.
A classic example of A/S collusion is an auctioneer that overstates the second highest bid to increase the seller's revenue. Another important kind of A/S collusion is represented by an auctioneer that declares no or a non-existent winning bidder due to too low bids.
An often neglected form of collusion is A/B collusion, e.g., an auctioneer that collaborates with the winning bidder and therefore intends to understate the selling price. However, in most real-world scenarios, auctioneers gain a fraction of the selling price and A/B collusions do not seem to be realistic. We therefore consider it as a minor form of collusion.

Collusions involving the auctioneer (A/S and A/B) are of particular interest in secure auction protocols because they allow agents to receive sensitive information from the auctioneer.

4 Publishing encrypted bids

As stated above, a bidder that won an auction cannot be sure whether the price the auctioneer tells him to pay is really the second highest bid. In the general case, he is even incapable of detecting whether the given price is one of the submitted bids, e.g., the auctioneer could make up a high bogus bid to increase the seller's revenue. A completely fake second-highest bid can be prevented by cryptographic signatures. However, this does not hinder a criminal auctioneer from having a bidder agent that submits a strategic bid *after* the auctioneer received and evaluated the other bids.

For this reason, it seems to be a good idea to divide the auction process into two parts. In the initial phase the auctioneer receives encrypted bids from the bidders and publishes the anonymized bids on a blackboard or via a multicast. The bids are encrypted by arbitrary personal keys created by each bidder. The auctioneer is not yet capable of opening the bids (**P1**).

The second phase starts after the submission deadline. As each bidder can observe the encrypted bids on the blackboard, the auctioneer is now unable to (undetectedly) alter existing or add fake bids. In the following, each bidder sends his key (masked with a random number to avoid bidder identification after the auction) to the auctioneer using public-key encryption. After having received the keys, the auctioneer secretly opens all bids, determines the second highest one and publishes the corresponding key. Additionally, he notifies the winning bidder and sets up the transaction.

This procedure voids the auctioneer's ability to change submitted bids or add bids after the submission deadline. Additionally, the auctioneer is incapable of overstating the selling price (**E2**). If the bidders are not collaborating with the auctioneer, they will detect the announcement of understated selling prices (**E3**). For instance, if the auctioneer deliberately declares the third highest bid as the selling price (to support the winning bidder), the second highest bidder observes this and can prove his claim by supplying his key.

However, if this bidder has no incentive to clarify the auction outcome, a collusion of the auctioneer and the two highest (or more) bidders ("A/B/B" collusion) can result in an undetectable understatement of the selling price. Another form of cheating in this protocol may occur when the auctioneer and a single bidder collude. The auctioneer can declare the highest bid as the selling price and his partner as the winner (violation of **E1**). However, the bidder has to pay more than he bid for the good, which he usually will not be willing to do. In an A/S collusion setting, the auctioneer could declare the highest bid as the selling price and secretly cancel the selling of the good, as all bidders assume another bidder won the auction. This can easily be prevented by publicly declaring the winner. All of the above problems (some of them are not relevant due to the rareness of A/B collusion) will be solved in the subsequent section.

As the protocol described is interactive, there is a problem when agents decide to inhibit the auction protocol by supplying no or false keys. We call this the *key denial* problem. It is impracticable to wait for each key forever because of **E6**. As a consequence, the auctioneer has to take measures if bidders are refusing to supply valid keys in time. Unfortunately, the key denial problem cannot simply be solved by withdrawing the undecryptable bids because a B/B collusion (that received private information from the auctioneer) can take advantage of this and understate the selling price. In order to assure a smooth auction process, uncooperative bidders have to be fined. As imposing sanctions for future auctions is not always feasible due to the anonymity in electronic marketplaces, a reasonable fine that allows easy implementation is to assign a *default bid* to incommoding bidders. In a setting where bidders receive confidential information from the auctioneer, there are two reasons for strategically denying a key: avoiding to win the auction due to an already opened second highest bid or manipulating the selling price after the highest bid has been opened. In the majority of auction environments, the default bid should be as high as possible. However, as the denial of a key will always falsify the actual auction outcome, a more rigorous measure is to fine the refusing bidder by voiding his bid and forcing him to pay a penalty. Obviously, this still does not meet the demands specified in **E4** and **E5**, but it can render the refusal of a key uneconomic.

5 Restriction of the auctioneer's knowledge

Given the procedure of the previous section, the auctioneer is unable to alter the outcome of the auction by making up fake bids. What remains to be done is to restrain his ability to release confidential information (bids and their origins) and to prove that the price he declares is actually the second highest bid. We will achieve this by the iterative opening of binary bidding lists. First of all, we discretize the bid interval into k possible bids $\{p_1, p_2 \ldots p_k\}$. Each bidder submits a bidding list that consists of k binary values denoting whether he is willing to pay a given price or not. Every bit in this list is encrypted using a different, arbitrary key K_{ij} generated by the bidder. These bidding lists are put together to the so-called *bid matrix* (see Table 1) and are published on a blackboard like in the previous section. As usual the functions $e(b, K)$ and $d(b, K)$ encode and decode a bid b with key K, respectively. To ensure anonymity of revealed bids the columns on the blackboard are presented in random order. The goal is to find an opening sequence that rapidly locates the second highest bid by revealing as little information as possible. Applying pre-defined sequence rules, the auctioneer requests key by key from the bidders until the second highest bid is undoubtedly detected.

	Bidder 1	Bidder 2	...	Bidder n
p_k	$e(b_{1k}, K_{1k})$	$e(b_{2k}, K_{2k})$	$\ldots$	$e(b_{nk}, K_{nk})$
p_{k-1}	$e(b_{1,k-1}, K_{1,k-1})$	$e(b_{2,k-1}, K_{2,k-1})$	$\ldots$	$e(b_{n,k-1}, K_{n,k-1})$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
p_1	$e(b_{11}, K_{11})$	$e(b_{21}, K_{21})$	$\ldots$	$e(b_{n1}, K_{n1})$

Table 1. Bid matrix

After having found the second highest bid p_y at position (x, y), the auctioneer publishes the *essential keys* E defined by the following equation.

$$E \quad = \quad \{K_{xy}\} \cup \{K_{i,\min(y+1,k)} \mid i \in \{1, 2 \ldots n\}\}$$

Figure 3 shows an example bid matrix ($n = 10$, $k = 15$, $p_1 = 5, p_2 = 10 \ldots p_{15} = 75$) and the keys to be published after the auction.

This proves to all participants that the declared selling price is actually the second highest bid. As a consequence, conditions **E2** and **E3** are fulfilled. Of course, keys for negative bids lower than p_{y+1} and for positive bids higher than p_{y+1} can be used as well, but this would give the bidders more information than necessary. It has to be decided as the case arises if requesting some additional keys can be afforded. As the columns of the bid matrix are shuffled, the bidders cannot relate bids and bidders.

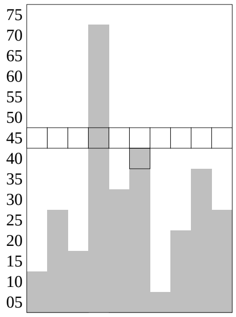


Fig. 3. Essential keys

In the following sections, we propose three different search procedures that locate and return the second highest bid. Each of them limits the auctioneer's knowledge of bids and their origins, resulting in a partial validity of **P2**.

Figure 4 formally summarizes the communication framework for the three procedures.

- ○ PHASE 1: Each bidder i supplies an ordered list of k encrypted bids (each one encrypted with an arbitrary key K_{ij}).
  — *Bid submission deadline* —
- – The auctioneer publishes the bidding lists on a blackboard (in random order).
- ○ PHASE 2: The following steps are repeated until the auctioneer has detected the second highest bid and (if desired) until he has received all essential keys (E).
  1. The auctioneer secretly[1] demands key K_{ij} from the corresponding bidder i (i and j are yielded by one of the algorithms in the subsequent sections).
  2. Bidder i sends K_{ij} and a random number encrypted with the auctioneer's public key.
     — *Key submission deadline* —
  3. The auctioneer verifies the key by attempting to decrypt b_{ij}. If the key is invalid or the bidder sent no key at all, a default bid is used and (if necessary) bidder i is fined.
- – The auctioneer publishes the keys in E (or another set of keys that proves the location of the second highest bid and the column of the winning bidder).
- – The seller and the winning bidder (who can identify himself by supplying the seller with all remaining keys of his column) get in contact and initiate the transaction.

[1] e.g., by using public key encryption.

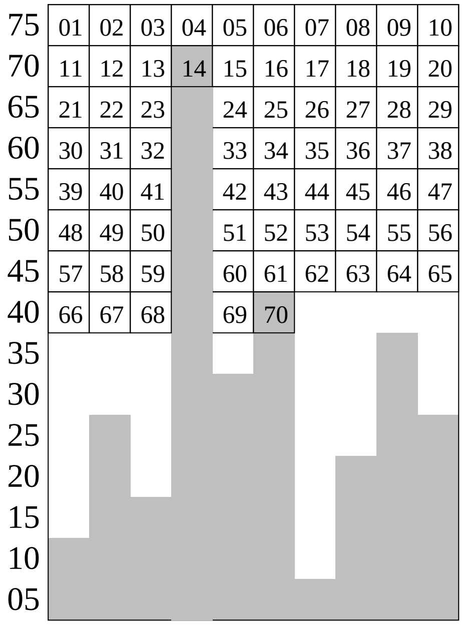
Fig. 4. Communication protocol

5.1 Downward bid search (`dbs`)

A straight-forward method to open the bids is to start at the highest price and open each row of bids downwards until at least two bidders are willing to pay a given price. This is similar to a *second-price* Dutch (descending) auction [22]. The following algorithm fulfills this task. To save space, we use $e_{ij} = e(b_{ij}, K_{ij})$ as an abbreviation for the encrypted bids and "$d(e_{ij}, K_{ij}) =$ true" if a bid is positive. The algorithm is decomposed into two separate procedures (`dbs` and `dbs2`) because we will reuse the second procedure for another search technique in a subsequent section. Decrypted bids are denoted by numbered frames in the example bid matrix, thus illustrating the opening sequence. The search begins in the upper left corner of the bid matrix by evaluating `dbs(1,k)`.

```
procedure int dbs($i$, $j$)
  while $j > 0$ do
    for $n$ times do
      if $d(e_{ij}, K_{ij}) =$ true then
        return dbs2($i$, $j$, $\{i\}$)
      end if
      $i = i + 1$
      if $i > n$ then $i = 1$ endif
    end for
    $j = j - 1$
  end while

procedure int dbs2($i$, $j$, $F$)
  while $j > 0$ do
    for $n$ times do
      if $i \notin F \wedge d(e_{ij}, K_{ij}) =$ true then
        return $j$
      end if
      $i = i + 1$
      if $i > n$ then $i = 1$ endif
    end for
    $j = j - 1$
  end while
```



75	01	02	03	04	05	06	07	08	09	10
70	11	12	13	14	15	16	17	18	19	20
65	21	22	23		24	25	26	27	28	29
60	30	31	32		33	34	35	36	37	38
55	39	40	41		42	43	44	45	46	47
50	48	49	50		51	52	53	54	55	56
45	57	58	59		60	61	62	63	64	65
40	66	67	68		69	70				
35										
30										
25										
20										
15										
10										
05										

The number of bids to open is $O(nk)$. After the decryption process, the auctioneer knows just two out of n bids (the highest and the mandatory second highest) and has no clue concerning the other bids. Furthermore, this search procedure has the advantage that the auctioneer is not capable of requesting more keys than he is allowed to (those beneath the declared selling price) without being detected afterwards. Please note that only the keys marked in Figure 3 are presented to the bidders.

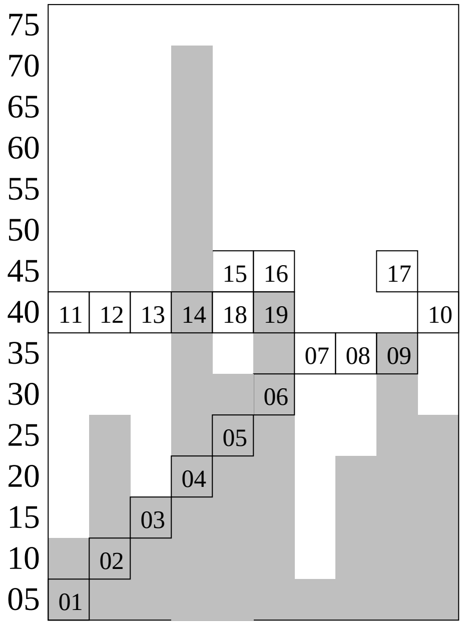
Although, revealing only one private value may seem a fairly good result, a disadvantage of this procedure is that unfortunately the highest bid usually requires the highest secrecy of all bids.

Bids that cannot be resolved due to denied keys should not be treated as default binary bids (neither negative nor positive) as this would enable easy price manipulation. Assigning a penalty of p_j for an undecryptable bid b_{ij} seems to be an appropriate measure. Bidders cannot take advantage of submitting *inconsistent lists*, i.e., lists that do not represent a single bidding value as only the first occurrence of a positive bid counts.

5.2 Upward bid search (ubs)

The following algorithm avoids the revelation of the highest bid by opening low bids first. When searching upwards, the auctioneer can skip to the next higher row when at least one bidder is willing to pay at a given price. The skipping of a line must not be triggered by the same bidder for two times consecutively. This technique resembles an ascending auction, in particular an English auction. The search starts in the lower left corner of the bid matrix (ubs(1,1)).

```
procedure int ubs($i, j$)
  $F = \emptyset$
  while $j \leq k$ do
    $p = 0$
    $F' = \emptyset$
    for $n - 1$ times do
      if $i \notin F$ then
        if $d(e_{ij}, K_{ij})$ = true then $p = 1$
        else $F' = F' \cup i$ endif
      end if
      $i = i + 1$
      if $i > n$ then $i = 1$ endif
      if $p = 1$ then break endif
    end for
    if $p = 0$ then break endif
    $j = j + 1$
    $i' = i$
    $F = F \cup F'$
  end while
  $i = i'$
  for $n - 1$ times do
    if $i \notin F \wedge d(e_{i,j-1}, K_{i,j-1})$ = true then
      return $j - 1$
    end if
    $i = i + 1$
    if $i > n$ then $i = 1$ endif
  end for
  return $j - 2$
```



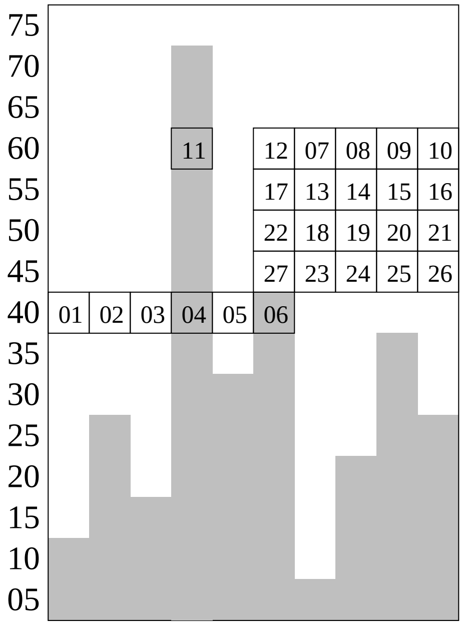
This algorithm is significantly faster than dbs ($O(\max(k, 2n))$). The auctioneer learns partial information about the losing bids and no information at all about the highest bid. This guarantees that the winning bid will remain secret even after the auction terminated. However, this type of bid searching may be unsuitable when $k \gg n$ because it reveals too much information about losing bids. The lowest bid can be determined to be in an interval of at most n bids. As bidders drop out when the price rises, the higher bids can be specified more precisely. The third highest bid is barely hidden from the auctioneer after ubs has been executed. It has to be one out of two possible values.

The key denial problem can be addressed more generously when searching upwards. It is sufficient for almost all scenarios to assess the value of denied bids as negative. It should be noted that due to the randomized order of the columns in the bid matrix, it is fair to open bids in linear order.

5.3 Binary bid search (`bbs`)

Related to well-known standard binary search, `bbs` begins in the middle of an interval by opening consecutive bids. After two positive bids have been found, the row is finished and `bbs` is called recursively for the upper half of the interval. If, after having opened all bids in a row, none of them is positive, the search is continued recursively in the lower half. If exactly one positive bid is found, `dbs2` is called from this point. `dbs2` reveals no additional information, except the required second highest bid. The search is initiated by executing `bbs(1,1,k,`$\emptyset$`)`.

```
procedure  int bbs($i, a, b, F$)
    $j = a + \lfloor \frac{b-a}{2} \rfloor$
    $p = 0$
    $F' = \emptyset$
    for $n$ times do
        if $i \notin F$ then
            if $d(e_{ij}, K_{ij}) =$ true then $p = p + 1$
            else $F' = F' \cup i$ endif
        end if
        if $p = 2$ then break endif
        $i = i + 1$
        if $i > n$ then $i = 1$ endif
    end for
    if $p = 2$ then return bbs($i, j, b, F \cup F'$) endif
    if $p = 0$ then return bbs($i, a, j, F$)
    else return dbs2($i, j + 1, F \cup i$) endif
```



This method seems to be a good compromise between both previous techniques. Partial information about some bids is revealed, but this information is by far not as precise as the one revealed by `ubs`. Because this algorithm uses `dbs2` to determine the second highest bid, it has the same round complexity as the downward bid search. Applying the binary search until the end would reduce the number of opened bids to $O(n \log_2(k))$, but this could reveal more information than needed. The search time can be further decreased by starting at the expected value of the second highest bid instead of the middle of the bid interval.

`bbs` is somewhat similar to the opening of bits in a standard binary representation of integers, but it has the advantage of uncovering less information.

It would be possible to execute this algorithm as an open-cry interactive auction. The auctioneer starts by asking the bidders in random order if they are willing to pay the price in the middle of the bidding interval. When two bidders accept the given price, he recursively performs this operation for the upper half of the interval. If all bidders reject the price, he continues with the lower half. In comparison with an open-exit English auction, this auction would have the advantages of lesser information revelation and faster execution time. However, the downward search (after the winner has been determined) had to be omitted as pricing could easily be manipulated by bidder collusions.

6 Conclusion

We presented a protocol that can be used in various ways to realize secure second-price auctions with a single auctioneer. Using a blackboard that displays encrypted binary bids, the auctioneer cannot alter the auction outcome without being detected. The needed trust in the auctioneer not to reveal private information can be vastly reduced by three search methods that restrict the auctioneer's knowledge in different ways. In fact, the auctioneer is reduced to a mediator that has no means to influence the result of an auction. It is even possible to completely omit the auctioneer and implement the suggested protocols in an open fashion that allows the bidders to resolve the auction on their own. This would equate all bidders since no-one could benefit from secret information received from the auctioneer. On the other hand, an auctioneer can be useful to *try* to conceal the sensitive data, at least. The bid opening protocol is interactive, but as it prohibits strategic interaction, agents are urged to supply all requested keys. Our technique meets the demands specified in **E1**, **E2**, **E3**, **E6** and **P1**. Practical compliance with **E4** and **E5** can be obtained by imposing fines to uncooperative bidders. `dbs`, `ubs` and `bbs` are three different ways to ensure the partial validity of **P2**.

Thinking of open networks, it would be necessary to carry out camouflage communication between the auctioneer and the bidders to hinder a bidder from drawing conclusions from the sequence of key requests. A drawback of our protocol obviously lies in its interactivity. The price determination might require lengthy communication between the auctioneer and bidders. However, in many real-world scenarios, the utmost secrecy of bids is more important than a rapid execution time.

Obviously, the discretization of the bid interval represents a limitation, but as bid values do not have to be equidistant, arbitrary bid sets, e.g., logarithmic scales can be used as well and enable an efficient partitioning of the bid interval.

In contrast to most existing second-price auction protocols, the proposed protocols do not have difficulties in detecting ties, i.e., more than one equal maximum bids.

The three suggested search techniques clearly illustrate the equivalence of Vickrey, second-price Dutch and English auctions for private value bidders. In addition, the binary search procedure demonstrates a novel method to locate the second highest bid. In the future, we intend to further investigate this new type of secure sealed-bid auction realization, implement, and evaluate the proposed procedures.

References

1. O. Baudron and J. Stern. Non-interactive private auctions. In *Pre-Proceedings of Financial Cryptography 2001*, pages 300–313, 2001.
2. F. Brandt, W. Brauer, and G. Weiß. Task assignment in multiagent systems based on Vickrey-type auctioning and leveled commitment contracting. In M. Klusch and L. Kerschberg, editors, *Cooperative Information Agents IV*, volume 1860 of *Lecture Notes in Artificial Intelligence*, pages 95–106, Berlin et al., 2000. Springer-Verlag.

3. 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.
4. K. Danielsen and M. Weiss. User control modes and IP allocation. http://www.press.umich.edu/jep/works/DanieContr.html, 1995. presented at MIT Workshop on Internet Economics.
5. K.E. Drexler and M.S. Miller. Incentive engineering for computational resource management. In B.A. Huberman, editor, *The Ecology of Computation*. The Netherlands, 1988.
6. J. Ferber. *Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence*. John Wiley & Sons Inc., New York, 1999.
7. M.K. Franklin and M.K. Reiter. The design and implementation of a secure auction service. *IEEE Trans. on Software Engineering*, 22(5):302–312, 1996.
8. M. Harkavy, J.D. Tygar, and H. Kikuchi. Electronic auctions with private bids. In *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pages 61–74, 1998.
9. B. Huberman and S.H. Clearwater. A multiagent system for controlling building environments. In *Proceedings of the 1st International Conference on Multiagent Systems (ICMAS-95)*, pages 171–176, Menlo Park, CA, 1995. AAAI Press.
10. M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts. In *Proceedings of Asiacrypt-00*, pages 162–177, 2000.
11. H. Kikuchi. (M+1)st-price auction protocol. In *Proceedings of Financial Cryptography (FC 2001)*, 2001.
12. 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.
13. M. Kudo. Secure electronic sealed-bid auction protocol with public key cryptography. *IEICE Trans. Fundamentals*, E81-A(1), 1998.
14. R.P. McAfee and J. McMillan. Auctions and Bidding. *Journal of Economic Literature*, 25:699–738, 1987.
15. P.R. Milgrom and R.J. Weber. A Theory of Auctions and Competitive Bidding. *Econometrica*, 50:1089–1122, 1982.
16. G.M.P. O'Hare and N.R. Jennings, editors. *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons Inc., New York, 1996.
17. M.H. Rothkopf and R.M. Harstad. Two models of bid-taker cheating in Vickrey auctions. *Journal of Business*, 68(2):257–267, 1995.
18. M.H. Rothkopf, T.J. Teisberg, and E.P. Kahn. Why are Vickrey auctions rare? *Journal of Political Economy*, 98(1):94–109, 1990.
19. T.W. Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In *Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS-96)*, Menlo Park, CA, 1996. AAAI Press.
20. A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
21. D.X. Song and J.K. Millen. Secure auctions in a publish/subscribe system. Available at http://www.csl.sri.com/users/millen/, 2000.
22. W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.
23. C.A. Waldspurger, T. Hogg, B. Huberman, J.O. Kephart, and W.S. Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117, 1992.
24. G. Weiß, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, 1999.
25. A.C. 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.