Improving Privacy in Vickrey Auctions

Natalia López, Manuel Núñez, Ismael Rodríguez, and Fernando Rubio
Universidad Complutense de Madrid

Many different types of auctions can be applied to determine selling prices, each of them fulfilling different properties. Among them, Vickrey auctions are specially interesting due to the fact that they disallow strategic behaviors of the bidders. In fact, the dominant strategy for each bidder consists in bidding his reserve price. However, somebody has to collect all the bids, so that bids are not kept private. In this paper we present a method to overcome this problem. That is, we present a way to implement Vickrey auctions preserving the privacy of all the bidders.

Categories and Subject Descriptors: K.4.4. **[Computing Milieux]**: COMPUTERS AND SOCIETY—*Electronic Commerce*

Additional Key Words and Phrases: Vickrey auctions, privacy

1. INTRODUCTION

Auctions have a special significance as economic mechanisms to determine allocation of resources, as well as the corresponding selling prices. Even though there are several variants, all of them follow a basic pattern. An auction is a market mechanism with an explicit set of rules determining resource allocation and prices on the basis of bids for market participants [McAfee and McMillan 1987]. Essentially, a group of *bidders*, which are the potential consumers of the auctioned item(s), submit *bids* to the *auctioneer*. Then, the auctioneer tries to choose the most favorable bid among all the available ones. Finally, the item(s) are assigned to the bidder(s) who submitted these *best* bids and the price is fixed.

Depending on how bids are submitted and on the final price, different types of auctions can be distinguished (see [Wurman et al. 1998] for a good taxonomy). Among the most popular auctions in the literature (see e.g. [Sandholm 1999; Vickrey 1961]) we may distinguish the following. In the *English auction* the bidders successively increment their bids until there is a bid that nobody increases. The item is assigned to the last bidder and the selling price is given by this last bid. In the *Dutch auction* the auctioneer starts with a (high) price and successively decreases it until a bidder accepts that price. Again, this bidder gets the item by paying the accepted price. In the *Sealed Bid auction* of First/Second price bidders submit their bids simultaneously without any information about other bidders preferences. The winner is given by the highest bid. In the case of *First* price

Author's address: Dept. Sistemas Informáticos y Programación, Universidad Complutense de Madrid, E-28040 Madrid. Spain.

the winner pays the highest bid (that is, his own bid). In contrast, in the case of *Second* price the bidder pays the second highest bid. This last type of auction is also known as *Vickrey auction* [Vickrey 1961].

As it is well-known, the Vickrey auction has several good properties. First, it removes any incentive for bidders to bid *strategically*. This is so because the dominant strategy of each agent consists in submitting a bid for his reserve price, that is, the maximum price that the agent would pay for the auctioned item. Thus, the Vickrey auction is a mechanism of *direct revelation* since, in order to maximize their utility, agents have to say the truth. Actually, the difference between the first and second prices is the *price* paid by the auctioneer to guarantee that all the agents tell the truth. However, as the *Revenue Equivalence Theorem* (RET) claims, all the four auctions presented before produce the same revenue for the auctioneer [Myerson 1981], though it is worth to point out that in general the auctioneer does not maximize the profit with respect to a more *flexible* scheme.[1] Actually, if the auctioneer would find out in advance the reserve price of the highest bid, he would prefer to sell the item with a fixed price as *take it or leave it*. Besides, the Vickrey auction is usually assumed to be a *private value* auction, that is, reserve prices are locally and independently fixed by each agent. This property disallows an agent to get more interested in an item because other agents have higher bids.

Even though nowadays most of the research in auctions is concentrated on more complex types of auctions (see e.g. the generalized Vickrey auction [Varian 1995] or the combinatorial auctions [Rothkopf et al. 1998; Parkes and Ungar 2000; Wurman and Wellman 2000; de Vries and Vohra 2003]) there is still room for improvements in single item auctions, in particular, concerning the Vickrey auction. First, given the fact that the auctioneer has access to all the bids, it may happen that he uses this information in subsequent auctions of similar items (by using a *take it or leave it* strategy). Thus, it is is not desirable for the agents that the auctioneer knows their reserve prices. Moreover, if the bidders know all the bids they can also adapt their subsequent bids.[2] This would imply that the auction is not with private value anymore, so that reserve prices are not used afterwards [Sandholm and Lesser 1995].

Thus, a desirable characteristic to be included in Vickrey auctions consists in keeping, as much as possible, the *privacy* of the bids. In other words, our goal is that at the end of the auction each bidder is the only one who knows his own bid. Moreover, it would be also very desirable that neither the bidders nor the auctioneer know the value of other bids. Obviously, there always exists some minimal exceptions to complete privacy. Actually, we need to know the second highest bid as well as the highest bidder. However, in order to resolve the auction we need to know neither the highest bid nor the second highest bidder.

The main goal of this paper is to provide a procedure so that auctions keep the good properties of Vickrey auction while privacy is also guaranteed, that is, bidders do not communicate their real bid to other agents (neither other bidders nor the

[1] The RET requires, for example, that the lowest bidder expects zero profit, that bidders are risk-neutral, and that bidders have independent and private values for their items. These properties are rarely met in real e-commerce environments.

[2] For instance, if a bidder played the role of an auctioneer in a previous auction where the item being sold was similar, then he could find out the bids in the current auction.

auctioneer). Let us remark that in our approach we will not depend on a trusted third part, as many cryptographic Vickrey auction schemes do (see e.g. [Naor et al. 1999; Lipmaa et al. 2002]). These schemes do not completely preserve privacy. For instance, in [Lipmaa et al. 2002] privacy is partially lost: although the *auction authority* cannot relate bids with bidders, he knows the value of all the bids that have been submitted. In the case of [Naor et al. 1999], the collusion of the auctioneer and the *auction issuer* allows them to infer all the bids of the bidders.

The rest of the paper is organized as follows. In the next section we sketch the main ideas of our method. Then, in Section 3 we introduce the main properties our auxiliary functions should hold. Afterwards, in Section 4 we present a concrete algorithm implementing Vickrey auctions that preserve the privacy of the bidders. For the sake of clarity, in Section 5 we present a small numerical example of our methodology. Finally, in Section 6 we present our conclusions and some lines for future work.

2. METHODOLOGY

In this section we review the major points underlying the implementation of our objectives. First, it is obvious that any approach based on Vickrey auction needs the value of the second highest bid (to fix the selling price). Thus, we need a mechanism to compare bids so that we can determine the second highest bid. Nevertheless, this has to be done without actually revealing the current bids. Intuitively, we will *transform* all the bids in such a way that we can compare these transformed values in the same way in which we would compare the original bids. Moreover, in order to assure privacy, we need to require that the transformation function is not *completely* known by any of the bidders nor by the auctioneer. Let us note that a total knowledge of the transformation function would allow a bidder to infer the original bids.

It is rather easy to get the first of the previous conditions (to keep the ordering between bids after transformation). Actually, it is enough to require the transformation function to be strictly increasing, that is, the considered functions f must fulfill $\forall\, x, y : x < y \Longrightarrow f(x) < f(y)$. In order to get the second condition (transformation function is unknown) we will provide a method so that this function is jointly defined by all of the bidders. In this case, each bidder will create a *part* of the function. Specifically, each bidder will be equipped with a *local function* $f : \mathbb{R} \to \mathbb{R}$ while the transformation function will be given by the composition of all the local functions. Afterwards, given a bid, we will consecutively apply each of the local functions to the value obtained previously. That is, bidders will apply their local functions to any value that they receive and then they will transmit the new value to another bidder. At the end of the process, that is, when all the original bids have passed through all the bidders, we get that the transformed bids have been constructed by the cooperative work of all the bidders.

In order to illustrate our strategy, based on the communication of values, we will present the (much simpler) following example. Let us suppose that we have n agents and that the agent i owns x_i units of money. We would like to know how many units, in average, they own. However, the personal richness of each agent should remain private. The solution is as follows. We arrange the agents from 1 to

n. The first agent communicates the value $x_1 + C$ to the second agent, being C a number generated by him (that is, nobody else knows this number). The second agent transmits to the third agent the previous value increased by his own number of units, and so on. That is, we have that the agent i communicates to the agent $i+1$ the value $C + \sum_{j=1}^{i} x_j$. At the end, when the first agent receives from the $n-th$ agent the value $C + \sum_{j=1}^{n} x_j$ he just subtracts C and divides by n.

We have already sketched the *high level* behavior of our algorithm, that is, agents will communicate values in order to encode together the original bids. However, we still have to study the properties that local functions must fulfill, since not any choice of local functions will allow us to reach the desired results. This is the objective of the next section.

3. PROPERTIES FOR LOCAL FUNCTIONS

As we indicated in the previous section, each bidder will privately define his local function. However, it will be necessary that he follows some *rules* previously agreed by all the bidders. That is, we will use conditions as "all the functions belong to the class $\mathcal{F}$". Next we analyze the sort of conditions that local functions and their application are required to fulfill.

3.1 Global transformation function is strictly increasing

As we have already commented, the global transformation function must be strictly increasing so that the relative ordering between bids is kept. A sufficient condition to assure that the transformation function is strictly increasing is that all the local functions are strictly increasing.

3.2 Each bidder applies his local function to his bid

We have to avoid situations where some of the bidders have *privileged* information about the original bids. For example, let us suppose that there exists a bidder being the first one to apply his local function to all the original bids. It is obvious that privacy is lost. In fact, as soon as one of the bidders transmits his bid to another one, privacy is already broken. The only possibility to avoid the previous problem is that the first step of the transformation is performed by each bidder to his own bid. That is, for each bid, the first local function applied to it is that of its bidder. Then, they may transmit the transformed value to the *next* bidder.

3.3 Composition of local functions is commutative

As a result of the condition explained before, we have that the order in which local functions are to be applied must be different for the different bids. The reason is not only that the first local function to be applied to each bid must be different; as we will see afterwards, the order of all of them must be different as well. However, we need that the overall composed function applied to all the bids is the same. Otherwise, the relative order of the original bids could be different to that of the transformed values of the bids.

In order to be sure that the composed function is the same independently of the order in which local functions are applied, it is enough to require that the composition of such local functions is commutative. That is, for every local functions f, g and for every value $x \in \mathbb{R}$ we must have $f(g(x)) = g(f(x))$.

3.4 Transforming the bids

Once all the bids have been transformed, they can be observed by all the bidders. Thus, it must not be possible for bidders to identify their own bids. Note that if a bidder identified his transformed bid, he would be able to obtain information about how many bids were higher than his own. In fact, if the order of application of local functions is predictable, each bidder could also know *who* have submitted bids higher than that of him. Thus, even if he cannot compute the exact value of such higher bids, part of the information will not remain as private as desired. Therefore, we should assure that our transforming mechanism guarantees that each bidder can track neither the transformations of his bid nor the transformation of any other bid. In order to do that, once he has applied his own local transformation, the order in which the rest of local functions are to be applied to the bid should be random. This can be done by communicating in each step not only the current state of the transformed bid, but also the set of bidders that have already applied their local functions to it. By doing so, at each step, the next bidder to apply its function will be chosen randomly from the complement of the set of bidders that have already applied theirs.

Let us remark that, as the composition of functions is required to be commutative, the order in which they are applied does not modify the overall result.

3.5 Recovering the second highest bid: Inverting the second highest transformed value

Once all the bids have been transformed, they can be trivially compared, and the second highest value can be obtained. Then, that (and only that) second value should be transformed back to its original value. This must be done to make public the price to be paid for the item. The process to obtain the real value will be simpler than the method used to codify them. It is enough to arrange the bidders in a random order, so that each of them applies the inverse of his local function to the value that it receives from its predecessor, and sends the resulting value to its successor. If the first element of the line receives the value to be decoded, the last element will compute the actual bid. Let us remark that the path used now can be different to that used while transforming the original bid. In fact, the last step can be done by any bidder, not only by the one submitting the bid being decoded.

After computing the real second highest bid, it will be enough to ask the bidders who is willing to pay more than that, and only the bidder with the highest bid will answer. Let us note that those bidders who offered less money will not be willing to pay that much. Moreover, the one who offered the second highest price will not care whether he obtain the item or not at that price, because he was using his reserve price. That is, he will not obtain any profit buying at his reserve price. These properties hold because we are using Vickrey auction, which guarantees that the dominant strategy consists in bidding the reserve price.[3]

Let us remark that in order to decode the bid it was necessary the collaboration of all the bidders, as no one could decode a value by his own. This fact preserved

[3]This statement could fail if certain RET assumptions do not hold. In particular, if the *values* are not *private*, then the reserve price of the bidder of the highest bid could change *after* he knows the second highest bid. In this case, he could *disown* his bid, and nobody would claim to be the highest bidder.

the privacy of the real bids. However, once one bid has been decoded, everybody knows an example of application of the composed function, as they know both the second highest bid and its codification. To guarantee privacy, the local functions must be selected in a way that the composed function cannot be inferred by a bidder even if he knows the output of the global function for a single input example.

4. CONCRETE DEFINITION OF LOCAL FUNCTIONS

According to the conditions stated so far, local functions should fulfill the following conditions:

(1) Their composition must be commutative.
(2) They must be strictly increasing.
(3) They must be such that the global function cannot be inferred from one single example of input and output.

There are many sets of functions fulfilling the previous conditions (see e.g. [López et al. 2003]). However, for the sake of clarity, we will slightly modify our approach in order to simplify the set of functions that we will use, in such a way that our privacy property will remain. Actually, this modification will allow us to use simpler sets of functions that do *not* fulfill all the three previous conditions. In order to present the proposed modification, let us consider some samples of very simple functions which are strictly increasing and whose composition is commutative. For instance, let us consider the set of functions of the form $f_k(x) = x + k$, and let us denote by α the set of all functions of this form. Trivially, we have that if $f_i, f_j \in \alpha$ then $f_i(f_j(x)) = f_j(f_i(x)) = x + i + j$, so the composition of two any functions of α is commutative. Besides, any function $f_i \in \alpha$ fulfills $\forall x, y : x < y \Rightarrow f_i(x) < f_i(y)$, that is, any function in the set is strictly increasing. Nevertheless, the set α does not fulfill condition (3), as one example of input and output allows to infer the global function. For instance, if we secretly choose two functions $f, g \in \alpha$ and we report that $f(g(2)) = g(f(2)) = 7$, then it is easy to infer that $f(g(x)) = x + 5$, even though neither f nor g are known. Therefore, if every bidder chooses secretly a function in α to be its local function then the privacy will not be preserved. This is also the case of some other sets of simple functions, like the set of functions of the form $f_k(x) = k \cdot x$ (namely β) or the set of functions of the form $f_k(x) = x^k$ (namely γ).

The problem in these cases is that these functions only depend on *one parameter* k. That is, there is only one unknown in the composed function. Thus, a single example provides an equation that can be used to infer the value of the unknown. Hence, we are interested in functions which depend on several parameters (for example, $f_{k_1,k_2}(x) = k_1 \cdot x + k_2$). Unfortunately, the composition of simple functions of this form is not commutative. So, the functions that need to be used to fulfill all the requirements are much more complex [López et al. 2003]. However, if we modify slightly our scheme then we can use simpler functions while keeping the desired properties. This will be done by performing the transformations of the bids in several successive *stages*. Let us remark that this modification will allow us to use functions such as those presented above, that is, functions where condition (3) does *not* hold. However, after combining several stages, condition (3) will hold, and so we will obtain the desired global function.

In the presentation made so far, we have assumed that the transformation is performed by applying consecutively the local functions of all the bidders. So, once all the bidders have applied their functions, the bid is completely transformed. On the contrary, from now on we will assume that the bid has to be transformed *twice or more times* by each bidder before it is completely transformed. Actually, the transformation will be split into various stages, so that in each stage all the bidders will apply once their local functions. The local functions to be applied by each bidder in each stage will be different, and the commutativity of the composition of the functions will only be required among the local functions applied in the *same* stage. Let us consider the following example with two stages. In the case of the first stage, the local functions to be used will belong to β, and in the second stage they will belong to α. So, once the first stage is finished, the composition of all the local functions of this stage will yield a global function of the form $f(x) = k \cdot x$. Then, the values obtained after the first stage will be introduced as inputs in the second. Thus, after all the local functions of the second stage are applied, the resulting transformation function computed from the original values will be of the form $g(x) = k \cdot x + r$. Obviously, this function depends on two parameters (k and r). Thus, a single data point (i.e. a single example of input and output) cannot allow to infer both parameters.

Nevertheless, the previous example is not completely valid. Recall that once all the values are transformed and compared, the transformation of the second highest value must be undone. That is, the original second highest bid must be recovered. This can be done by going back through the second stage and then through the first stage by applying the inverse local functions. Let us consider the bidder B who applies the last inverse local function of the second stage. After B applies its inverse function, he transmits the resulting value to any other bidder B' to begin the inverse transformation of the first stage. Then, contrarily to all the other bidders, bidders B and B' will know not only the original bid and the transformed value after the two stages, but also the intermediate transformed value after the first stage and before the second. Thus, they will have a system with two equations and two unknowns, that can be trivially solved. Hence, bidders B and B' can infer the complete transformation function, so that the original bids are not private any more.

Obviously, we can avoid the previous problem by using *three* stages instead of two. In that case, there will be three unknowns, so that three equations will be needed to infer the composed transformation function. However, care need to be taken to guarantee that no bidder knows both of the two intermediate values computed after undoing stages three and two. Let us denote by S_1 the set of bidders that participate in the middle of stages one and two, and let S_2 be the set of bidders in the middle of stages two and three.[4] In order to guarantee that no bidder can infer the complete transformation function, it is enough to require that $S_1 \bigcap S_2 = \emptyset$. In this case, a bidder in S_1 will be able to infer the global function of stage one, as he will know an example of input and output of this stage. However, he will be unable to infer the global functions of stages two and three, nor the composition of

[4]Let us remark that the number of elements in any of these sets must be 1 or 2. It will be 1 if the last bidder of a stage and the first in the following stage is the same bidder.

the three stages. The case is similar for bidders in set S_2.

Let us make some remarks regarding the use of several transformation stages. First, let us note that we could apply more than three stages. By using more stages we will reduce the relevance of the data each bidder owns, as we will increment the number of unknowns, but we will not increment the number of equations each bidder knows. Second, it is easy to guarantee that each bidder appears at most once in the middle of two stages while applying the inverse functions: we can do it in the same way used to guarantee that each bidder is used once in each stage, that is, using a set of already used bidders. Thus, the data each bidder will transmit to the next bidder will consist of three elements: the value of a bid partially transformed, the set of bidders which have already applied their local functions to this value in the current stage, and the set of bidders which have already been used in the middle of two of the previous stages. The former set, which will be used in both the encoding of all bids and the decoding of the second highest bid, is empty at the beginning of the encoding/decoding of every new stage, while the latter, that will be used only in the decoding, is only empty at the beginning of the last stage (i.e., at the first stage to be decoded). Third, let us note that the families of functions used in two consecutive stages should be different. It is worth to point out that in the other case they could collapse into a single stage, making it possible to infer the composition of their global functions. In order to make consecutive families different, it is enough to use sets α and β alternatively through the transformation.

For the sake of clarifying the previous concepts, a simple numerical example of the application of our procedure is presented in the next section. Besides, the distributed algorithm performing our method and taking into account all the previous considerations is depicted in Figure 1.

Finally, let us remark that by using at least three stages we also eliminate the problem of *collusion* of bidders. A relevant issue that must be addressed in our auction scheme is whether the collusion of two or more bidders could allow them to break the privacy. The first difficulty for bidders to collude in our scheme lies in the fact that each bidder does not know who to be in collusion with: The only way to know it is to ask everybody. However, in the worst case a bidder could use two different bidders in the same auction (one of the bidding 0) in order to be in collusion with himself. Even in this case he should be very lucky to get both agents positioned in the intermediate points of stages. Nevertheless, even this risk can be eliminated. Actually, if the number of stages is equal to the number of bidders, then all the bidders have a different portion of the information needed to infer the transformation function. Moreover, in this case the only way a bidder has to infer the function is to collude with *all* of the other bidders to exchange their own information. Obviously, there exists no mechanism to avoid collusion of bidders when *all of them* agree to break the privacy, so we can ignore this case. Let us note that applying a number of stages equal to the number of bidders makes the complexity of our algorithm to be in $\mathcal{O}(n^2)$, while its complexity is in $\mathcal{O}(n)$ when the number of stages is constant. Hence, the collusion risks must be considered prior to fix the number of stages.

Notation:

—P: set of bidders, with $n = |P|$.

—Given $p_i \in P$, the bid of p_i is b_i.

—ST: number of stages.

—S: set of bidders already used in the current stage.

—G: set of bidders already used in intermediate points (in the inverse transformation).

Initialization:

(1) ST is agreed by all bidders ($ST \geq 3$).

(2) The set of functions π_i of each stage i is fixed commonly by all bidders such that:
—$\forall\ 1 \leq i \leq ST-1 : \pi_i \neq \pi_{i+1}$
—$\forall\ 1 \leq i \leq ST : \pi_i \in \{\alpha, \beta\}$ where $\alpha = \{f \mid f(x) = x + l\}$ and $\beta = \{f \mid f(x) = l \cdot x\}$

(3) For each bidder $p_j \in P$, privately do:
(a) For each stage i, choose a function $f_{ji} \in \pi_i$.
(b) $c := f_{j1}(b_j)$
(c) $p := ChooseRandomly(P \backslash \{p_j\})$
(d) Transmit $(c, \{p_j\}, 1)$ to bidder p /* *where 1 denotes the first stage* */

Inductive Case (forward way): When tuple (c, S, i) is transmitted to bidder p_j, do:

(1) $c := f_{ji}(c)$

(2) **If** $|S| \leq n-2$ **then** /* *more bidders must apply their functions* */
(a) $p := ChooseRandomly(P \backslash (S \cup \{p_j\}))$
(b) Transmit $(c, S \cup \{p_j\}, i)$ to bidder p
else if $i < ST$ **then** /* *there are more stages* */
(a) $p := ChooseRandomly(P)$
(b) Transmit $(c, \emptyset, i+1)$ to bidder p
else Broadcast c

Comparison: Once n values have been broadcasted, we publicly perform:

(1) Obtain the second highest value (namely d).

(2) $p := ChooseRandomly(P)$; $G := \emptyset$

(3) Transmit $(d, \emptyset, ST, G)$ to bidder p

Inductive Case (backward way): When tuple (c, S, i, G) is transmitted to bidder p_j, do:

(1) $c := f_{ji}^{-1}(c)$

(2) **If** $|S| \leq n-2$ **then** /* *more bidders must apply their functions* */
(a) $p := MaybeChooseRandomly((P \backslash (S \cup \{p_j\})) \cap G)$
(b) **If** p does not exist **then** $p := ChooseRandomly(P \backslash (S \cup \{p_j\}))$
/* *Depending on whether p is the last bidder of this stage:* */
(c) **If** $|S| \leq n-3$ **then** Transmit $(c, S \cup \{p_j\}, i, G)$ to bidder p
else Transmit $(c, S \cup \{p_j\}, i, G \cup \{p\})$ to bidder p
else if $i > 1$ **then** /* *there are more stages* */
(a) $p := ChooseRandomly(P \backslash G)$
(b) Transmit $(c, \emptyset, i-1, G \cup \{p\})$ to bidder p /*p *is first bidder of next stage*/
else Broadcast c

Resolution: Once a value c is broadcasted, do:

(1) Ask for the bidder whose bid was greater than c. Let k be such bidder.

(2) Assign item to bidder k at price c.

Fig. 1. Actual algorithm.

5. SMALL NUMERICAL EXAMPLE

In this section we present a small numerical example to illustrate the algorithm described in the previous section. Let us suppose that we want to perform an auction with bidders B_1, B_2, B_3, B_4. Besides, we will perform the codification of bids in four stages. For the sake of variety, we will use the three kind of functions commented in the previous section, that is, α, β, and γ (which denote functions of the form $f(x) = x + k$, $f(x) = k \cdot x$, and $f(x) = x^k$, respectively). In the first and the last stages functions in β will be applied. In the second stage functions will belong to α. Besides, in the third stage we will take our functions from γ. For the sake of clarity, we will assign small values to the constants k in each function.

Let us consider the path of transformations we must perform to encode the bid $b_1 = 10$ of bidder B_1. The transformation begins at stage 1. Let us suppose that $k = 3$ for bidder B_1 (that is, function of B_1 for stage 1 is $f(x) = 3 \cdot x$). Besides, let us suppose that the value of k for bidders B_2, B_3, and B_4 is $\frac{1}{2}$, 4, and $\frac{1}{5}$, respectively. Finally, let us suppose that the transformation of b_1 is performed in the following order: B_1, B_2, B_3, B_4. Then, B_1 transmits 30 to B_2, B_2 sends 15 to B_3, B_3 transmits 60 to B_4, and B_4 finishes stage 1 after obtaining 12. Let us note that each of these values is known only by the corresponding sender and receiver.

Now, stage 2 begins. Let us suppose that values of k are 5, 13, 7, and 20 for B_1, B_2, B_3, and B_4. Besides, let us suppose that this time the order of bidders is B_4, B_2, B_3, B_1. After B_1 applies his function, the new value is $12+(20+13+7+5) = 12 + 45 = 57$.

Then, stage 3 is performed. This time, k-values are 2, 1, 3, and 1, respectively. Besides, the order of bidders is B_1, B_4, B_2, B_3. The value after stage 3 is $57^6 = 34296447249$. Let us note that by using functions from set γ the transformed values could be extremely high, even in environments with not many bidders.

Finally, stage 4 is performed. The k-values are now 2, 7, 4, and $\frac{1}{2}$, and this time the order is B_3, B_4, B_1, B_2. The final value after the whole transformation is $34296447249 \cdot 28 = 960300522972$. After B_2 obtains that value, he broadcasts it to all bidders.

The previous procedure will also be applied to bids b_2, b_3, and b_4 belonging to bidders B_2, B_3, B_4, respectively. In fact, in all cases the same global function $f(x) = (((\frac{6}{5} \cdot x) + 45)^6) \cdot 28$ will be applied (but the order of application of local functions in each of the four stages might be different for each bid). The relative order of bids is preserved after the application of the function, so we can compare the codified bids to identify the second highest bid. Besides, let us note that no bidder knows which of the four resulting codes matches his own bid, as the path of function applications is random. So, the actual bidder that broadcasts each encoded bid after the last stage finishes provides no clue to guess the identity of the owner of such a bid.

Let us suppose that after comparing the four encoded values we notice that b_1 is the second highest bid. Then, the value 960300522972 (and only this value) is decoded by performing the reverse functions from stage 4 to stage 1. Let us suppose that this reverse transformation is performed in the following order: $(B_3, B_1, B_4, B_2), (B_2, B_4, B_3, B_1), (B_1, B_4, B_2, B_3), (B_3, B_1, B_3, B_4)$. After the transformation of this value is undone, the bid 10 is broadcasted. Let us consider the

information of each bidder at this point. Let us remark that all bidders know the second highest value (10) and its corresponding transformation (960300522972), as both values are publicly reported. Besides, bidder B_3 knows the value 34296447249 of the bid after stages 1, 2, and 3, because he applied the last local function of stage 3 and the first of stage 4. Hence, he can easily infer that the global factor in stage 4 is 28, because $34296447249 \cdot 28 = 960300522972$. Besides, he knows that $((a \cdot 10) + b)^c = 34296447249$, but he has an equation and three unknowns. Hence, he cannot infer the global function f. A similar argument applies to B_2: He can infer that the multiplicative factor in stage 1 is $\frac{6}{5}$, but the equation with 3 unknowns $((\frac{6}{5} \cdot 10 + b)^c) \cdot d = 960300522972$ cannot be solved by him. Following the same ideas, B_1 has 2 equations each one of two unknowns, but the unknowns of one equation are disjoint from the unknowns of the other.

At this point, the unique bidder whose bid was over 10 will claim to be the auction winner, and the item will be sold to him. Let us note that due to the Vickrey auction properties he is also the unique bidder willing to pay more than 10, as the reserve price of the second bidder is exactly 10.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a mechanism to perform the Vickrey auction so that only the minimal information needed to resolve it is made public. That is, the only data publicly known are the second highest bid and the identity of the bidder who made the highest bid. Actually, any other relevant data (bids different from the second, identities of bidders different from the first) are kept private by their legitimate owners. These data are never communicated to any other bidder, nor to the auctioneer, nor even to a third party in which we must trust. So, they remain completely secret in the machines of their respective owners (assuming that nobody but the owner can access the data stored in each machine). Therefore, our mechanism provides a way to perform private Vickrey auctions such that we need neither any third party nor even the auctioneer.

Our algorithm is based on resolving the auction by comparing not the original bids but the bids transformed according to some transformation function. This function is built collaboratively by all the bidders in such a way that no bidder knows it completely. The function cannot be inferred by any participant, because the only sample of input and output which is communicated (that of the second highest bid) is not enough to find it out, as it depends on several degrees of freedom. So, the transformed bids do not provide any relevant data about the original bids: Neither their values nor the difference between each pair of transformed bids are significant.

Let us note that our current algorithm requires the bidders to somehow trust each other. This is so because, although it is impossible that a bidder infers the bid of other bidder, it is possible that a bidder does not apply correctly his own functions. By doing so, the computation of the second highest bid could be incorrect. Some manipulations are easily detectable: If false data make the relative place of first or second bidder to change, we could have at the end that no bidder or more than one bidder claim to be the first bidder. Nevertheless, a manipulation of the second price with no order modification is more difficult to detect. We are currently working on

a kind of *checksum* method that can detect whether somebody has lied or not. In fact, we can currently detect lies, although the method cannot always detect who was the liar.

Acknowledgments

This work has been supported in part by the MCYT project MASTER (TIC2003-07848-C02-01), and by the JCCLM project (PAC-03-001). Besides, we would like to thank the anonymous referees of this paper for their suggestions and helpful comments.

REFERENCES

de Vries, S. and Vohra, R. 2003. Combinatorial auctions: A survey. *INFORMS Journal on Computing 15,* 3, 284–309.

Lipmaa, H., Asokan, N., and Niemi, V. 2002. Secure Vickrey auctions without threshold trust. In *Annual Conference on Financial Cryptography, LNCS 2357*. Springer, 87–101.

López, N., Nuñez, M., Rodríguez, I., and Rubio, F. 2003. Technical quote: Functions fulfilling conditions (1), (2), and (3). `http://dalila.sip.ucm.es/~fernando/TechQuote.ps`.

McAfee, R. and McMillan, J. 1987. Auctions and bidding. *Journal of Economic Literature XXV*, 699–738.

Myerson, R. 1981. Optimal auction design. *Mathematics of Operations Research 6*, 58–73.

Naor, M., Pinkas, B., and Sumner, R. 1999. Privacy preserving auctions and mechanism design. In *ACM Conference on Electronic Commerce*. ACM Press, 129–139.

Parkes, D. and Ungar, L. 2000. Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*. 74–81.

Rothkopf, M., Pekec, A., and Harstad, R. 1998. Computationally manageable combinatorial auctions. *Management Sci. 44,* 9, 1131–1147.

Sandholm, T. 1999. Distributed rational decision making. In *Multiagent Systems*. The MIT Press, 201–258.

Sandholm, T. and Lesser, V. 1995. On automated contracting in multi-enterprise manufacturing. In *Distributed Enterprise: Advanced Systems and Tools*. 33–42.

Varian, H. 1995. Economic mechanism design for computerized agents. In *USENIX Workshop on Electronic Commerce*.

Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance 16*, 8–37.

Wurman, P. and Wellman, M. 2000. Akba: A progressive, anonymous-price combinatorial auction. In *Second ACM Conference on Electronic Commerce*. ACM, 21–29.

Wurman, P., Wellman, M., and Walsh, W. 1998. The Michigan Internet Auctionbot: A configuration auction server for human and software agents. In *Proceedings of the Second International Conference on Autonomous Agents*. ACM Press, 301–308.