Proceedings of the 3rd USENIX Workshop on Electronic Commerce, September 1998, pp. 61-73

Electronic Auctions with Private Bids

Michael Harkavy
Carnegie Mellon University
Pittsburgh, PA 15213
bif@cs.cmu.edu

J. D. Tygar
Carnegie Mellon University
Pittsburgh, PA 15213
tygar@cs.cmu.edu

Hiroaki Kikuchi
Tokai University
Hiratsuka, Kanagawa, Japan
kikn@ep.u-tokai.ac.jp

Abstract

Auctions are a fundamental electronic commerce technology. We describe a set of protocols for performing sealed-bid electronic auctions which preserve the privacy of the submitted bids using a form of secure distributed computation. Bids are never revealed to any party, even after the auction is completed. Both first-price and second-price (Vickrey) auctions are supported, and the computational costs of the methods are low enough to allow their use in many real-world auction situations.

1 Introduction

Auctions are a fundamental technology for electronic commerce. They have been suggested as a technology for controlling allocation of bandwidth [5, 10] and are increasingly seen on the web.

If we could have an ideal auction, what properties might we desire? Here are some desiderata (not a complete list) for an ideal auction:

- **Economic design** — we want the auction to be designed on solid economic principles and for participants to have incentives to bid as they truly value the item — this is known as their *valuation*, and is also called their *indfference price*. If bidders bid less than their true valuations, it is possible that the final winning bid may be artificially low — we illustrate this below in our discussion of sealed-bid auctions.
- **Fast execution** — we want to have the auction run quickly.
- **Privacy** — we want the auction to be private — for others to not know our actual bids. We do not want even an auctioneer to know the bids. The only exception to this rule is that we will reveal the final price at which the item is sold. (At first this may seem like a paradoxical condition, but it is commonly achieved in Dutch auctions, discussed below.) Note that this is a quite useful requirement — otherwise we give away detailed information on our preferences that may be used in the future to inform "shills" who work for the seller to attempt to artificially drive up the price an item is sold at (creating a disincentive to bid the true valuation.)
- **Anonymity** — we don't want our identities to be revealed. One way to achieve this is to use an intermediary to anonymously forward our bids. Note that privacy is different from anonymity; privacy protects the values of the bids while anonymity protects the identities of the bidders. Even if our bids are anonymously forwarded, participants (such as the auctioneer) may learn the distribution of our bids.

In this paper, we discuss how to hold a true auction that combines the first three features. If we add anonymizing intermediaries to the mix, we can achieve an auction with all four properties.

1.1 Auction Types

How do existing auction types stack up against our desiderata?

Consider these three broad categories of auctions that have been proposed:

- *Increasing-price auction (English auction).* In this type of auction, a good or commodity is offered at increasing prices. It may initially be offered at K tokens; at successive points of time i it is bid at $K + i * \Delta$ tokens (Δ may be a function of previous bids and other factors). At each unit of time, one or more parties

As of September 1, 1998, the first two authors can be reached at Computer Science Division, University of California, Berkeley, CA 94720. Sponsored by DARPA under grant F19628-96-C-0061, the U.S. Postal Service, and Toshiba Corporation. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation thereon. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official poilicies, either express or implied, of any of the supporting organizations or the U.S Government.

can bid for the item. At the end of the auction, the highest bidder takes the item; he pays the price he bid. This is the sort of auction found at Sotheby's and Christie's. This type of auction has many disadvantages: the time necessary to conduct the auction is potentially proportional to the price at which the item is sold; the communication costs may grow superlinearly in the ultimate price at which the item is sold (since at lower prices, multiple bidders may simultaneously bid for an item); moreover, this type of auction leaks an enormous amount of information—a careful observer will be able to deduce information about the price that each party is willing to pay for the auctioned good. However, the auction does have a very desirable feature: in economic terms, it allocates the good to the bidder with the highest valuation, since the bidder with the highest valuation will be willing to outbid all other bidders.

- *Sealed-bid auctions.* In this type of auction, each party sends a sealed bid to an auctioneer who opens all bids. The auctioneer determines the highest bid and sells the item to that bidder for the bidding price. This type of auction can execute in a single round of communication between the bidders and the auctioneers. However, it has disadvantages. First, the auctioneer will know the exact price that each party is willing to pay. Second, it does not support optimal distribution of goods.

  In a sealed bid auction, participants will have beliefs about what others will bid. If a participant believes that she will have the highest bid, and the second highest bid will be substantially beneath that, then she has an incentive to lower her bid. For example, if she values an item at \$1,000, but believes that the second highest bidder values the item at \$500, then she is likely to place a bid slightly higher than \$500. If she is wrong about the distribution of other bids, then the final item will not go to the party that values it most, and the seller will have given up the item a price lower than he would have achieved with an English auction. [9, 11, 13, 14].

- *Decreasing-price auction (Dutch auction).* This type of auction is similar to the English auction in that the bidding price varies over time; however, in this case, the price decreases and at time i is $K - i * \Delta$. The first bidder will take the item. This type of auction has the advantage of preserving maximum privacy — no information is revealed except the winning bid and bidder; however like the increasing-price auction, it may be time consuming, and like the sealed-bid auction, it is not economically efficient [9, 11, 13, 14].

In Nobel-prize winning work, the economist Vickrey designed a type of auction that combined the best features of an increasing-price auction and a sealed-bid auction [13]. Vickrey's technique, called a *second-price auction*, works like a sealed-bid auction, in that all bids are sealed and sent to an auctioneer. Like a sealed bid auction, the highest bidder wins. But the price the winner pays is the price that the second highest bidder has bid. For example, suppose that we bid 100 tokens and the second highest bid is 10 tokens. Then we will win the bid, but we will only have to pay 10 tokens to secure the good. This auction runs in constant time, and maximizes consumer surplus, but it is still highly centralized and does not protect the privacy of the bids.

The main contribution of this paper is to give a private-bid version of a second-price auction. This auction

- will run in a single round of bid submissions (like a sealed-bid auction),
- is efficient enough for practical implementation,
- will maximize consumer surplus and will give incentives for participants to submit bids at their true valuations (like an English auction), and
- will preserve bid privacy (like a Dutch auction).

This is quite an unusual result. In the end, only the second highest bid is revealed — the auctioneers and participants (except for the winner) will be completely unaware of the numerical value of the highest bid (or any other bid besides the second highest).

1.2 Secret computation

Our approach builds on a long tradition of secret function computations. Researchers have developed a number of techniques for securely computing arbitrary functions—some major examples are [2, 8, 3]. This means that given a set of participants, each with an input, we can compute (with a polynomial slowdown) any function of the inputs; and moreover, we can do so with a protocol that leaks no information to any participant. The only information that a participant will know about the the input of other particpants is information derivable from his own input and the final result.

Unfortunately, the general techniques for secure computation are impractical for general use. They do, in fact, run in polynomial time, but with a big explosion of states. Thus, the development of private protocols for specific problems must be done by hand. We have developed and tuned our protocol to run quickly — and we believe that it is practical for real use. We plan to build a prototype system embedding this algorithm to prove its practicality.

In the remaining sections, we first consider a simple private sealed-bid auction; that is, we show how to compute the *max* function privately. Then, we describe a fully private and secure second-price auction protocol which is the result of several improvements on the original.

The essence of the second-price protocol is a series of computations which securely determine whether or not there are at least two bids greater than or equal to a specific value. This test is performed by determining if there is some partition of the set $\mathcal{B}$ of n bidders into $\mathcal{B}_1$ and $\mathcal{B}_2$ such that there is a bid at least as high as the test value on each side of the partition. We need not actually consider all possible partitions; we can select a small number ($\log_2 n$) of partitions which will suffice. We iterate this series of computations in a search for the value of the second highest bid.

2 A simple auction method

First, we will briefly describe a simple first-price auction protocol with private bids which captures some of the flavor of the more complex version to follow. We assume that all bids are drawn from some ordered set $\Delta = \{\delta_1, \ldots, \delta_V\}$, and that there are n bidders and m auctioneers. We also fix a sufficiently large (64–128 bit) prime number p. All arithmetic will be computed modulo p. Each bidder composes her bid by creating V lists of m integers modulo p. The rule for creating these sets is that:

- If the bidder is willing to pay δ_l for the object, then the l^{th} list of numbers is chosen randomly with uniform distribution subject to the constraint that the sum (modulo p) of all numbers must be non-zero.
- If the bidder is unwilling to pay δ_l for the object, then the l^{th} list of numbers is chosen randomly with uniform distribution subject to the constraint that the sum (modulo p) of all numbers must be zero.

Each auctioneer is sent V numbers by each bidder; the i^{th} auctioneer is sent the i^{th} number from each list. The submissions are signed by the bidders, and signed receipts are issued by the auctioneers. The signature by the bidder should be a signature on a hash of hashes of each of the individual values in the list so that the signature can be used to prove a bid at one particular value without revealing the bids at any other values. Strange or non-random bidding behaviors are possible (e.g. being willing to pay δ_j but not δ_i for $i < j$). Such inconsistencies could be easily eliminated by methods similar to those outlined in section 4.1.1.

To determine the winning bid, the auctioneers compute, for each δ_l, the sum of the m (one from each bidder) numbers received for that δ_l. To determine whether a particular δ_l is above or below the winning bid, the auctioneers commit to and reveal their sums for that δ_l, and then compute a sum of all these values (which is equal to the sum of all numbers from all bidders for δ_l). The sum of all values for a particular δ_l is non-zero (with high probability) exactly when at least one bidder is willing to pay δ_l. The auctioneers find the highest l for which some bidder is willing to pay δ_l, possibly in several stages with a branching search for the highest bid. Once the highest bid is known, the auctioneers can then use the signatures on the bid submissions to prove which bidder is the winner by reassembling all bids for that winning δ_{l_0}.

This auction has some weaknesses. Most significantly, a coalition of an auctioneer and the highest bidder might uncover some information about the second highest bid. This information would be leaked by the fact that, for all l lower than the highest bid but higher than the second highest bid, the total of all bids would be simply the bid of the highest bidder. The obvious way to avoid revealing these intermediate sums would be to reveal sums one at a time from δ_V down and would require a number of rounds linear in the number of bidding points, which is likely to be unacceptably slow. Another weakness is that the work required (both computation and communication) scales linearly with the number of bidding points. For high value auctions, the desirable number of bidding points might be quite large.

We will give an auction in which the work scales only logarithmically with the number of bidding points, which provides complete privacy, and which allows for second-price auctions. It is worth noting that these alterations need not be combined; any subset may be implemented and computation costs will vary accordingly.

3 Secure distributed computation

This section provides an overview of the methods used as the computational basis for the protocol, which come from Ben-Or, Goldwasser, and Wigderson[2]. For further details or proofs, we refer the reader to this work and its extension by Beaver[1]. Information is distributed among the agents (the auctioneers in our case) by means of polynomials over a finite field as introduced by Shamir[12], but with verifiability as in [2, 4]. These polynomials are stored in a point-value representation, where each of the m agents has a specific point $\alpha_i \neq 0$ in the field at which he knows the value of all shared polynomials. The value of a polynomial at a specific point is known as a share. The initial polynomials will have some fixed degree-t. Since higher-order coefficients for these polynomials will be randomly selected from a set including zero, it might be more precise to say "degree bound t", but this distinction is never relevant to our results and we will freely ignore it.

Individual pieces of data are represented as points in the finite field and encoded in a shared polynomial as the free coefficient of that polynomial (which is the same as the evaluation of the polynomial at the point 0). All other coefficients are chosen from the field uniformly at random. Over a finite field (e.g. Z_p for prime p), any set of t known values of a degree-t polynomial (chosen as described) gives information-theoretically zero information about the polynomial's value at any other point. This property that any t shares of a secret yield no information about the secret's value is known as t-privacy. There is a (simple and short) interactive protocol which ensures that the set of shares held by correct agents represents a degree-t polynomial.

The data used in the computation needs to be not only private, but tamper-proof as well. This property is supported by a special selection of the evaluation points. The α_i are chosen to be the set of powers of a primitive m^{th} root of unity (say ω). That ω is a primitive m^{th} root means $\omega^m = 1$ and for $i : 1 \leq i < m$, $\omega^i \neq 1$. Obviously we must choose a finite field which has such a primitive m^{th} root, but this is not difficult (m is small). Given this choice of α_i, the shares of the polynomial are an error correcting code. Up to $\lfloor \frac{m-1}{3} \rfloor$ shares may be missing or incorrect and recovery of the free coefficient will still be possible. The resistance to up to t false values is known as t-resilience.

The basic computational operations are the addition or multiplication of two polynomials. The value of a sum or product of two polynomials at a point is the sum or product of the values of the two polynomials at that point. So, these operations are done pointwise, with each agent simply computing the addition or multiplication of the values at the point held. For addition (or multiplication by a scalar), this process is complete, privacy is not interfered with, and no other work needs to be performed. Extra work is required for multiplication.

The difficulty with multiplication is that the degree of the product of two polynomials is the sum of their degrees. The number of points needed to specify a degree-t polynomial is $t+1$, while to specify the product of two degree-t polynomials requires $2t+1$ points. Repeated multiplications will continue to increase the degree of the shared polynomials. Since the polynomials are shared among a fixed number of agents, the degree of the polynomial will rapidly exceed the ability of the agents to represent it. Repeated multiplication requires some means of reducing the degree of a polynomial without changing or revealing information about its free coefficient. A related difficulty is that the product of two polynomials does not have the randomness property which we required to enforce privacy. Many polynomials of degree $2t$ can not be expressed as the product of two degree-t polynomials (e.g. irreducible polynomials).

These two difficulties are solved together by a stage of communication between the agents which produces a degree-t polynomial whose free coefficient is the same as the free coefficient of the product of 2 shared degree-t polynomials and whose remaining coefficients are uniformly random.

The key observation (which we state without proof) is as follows. If we interpret the m shares of a polynomial as a vector, there is a constant $m \times m$ matrix (based on the the α_i) which transforms the shares of any degree-$2t$ polynomial into shares of a degree-t polynomial while leaving the $t+1$ lowest-order coefficients unchanged. Each agent can compute a vector of values (call it V_i) by multiplying its share of the polynomial by a vector of constants (corresponding to row i of the matrix). Now $\sum_i V_i$ is the vector of shares of the transformed polynomial. In order to mask non-uniformity of the coefficients of the product of polynomials, a random degree-t polynomials is generated and added to the vector components before they are shared. The agents sum all received values to compute their shares of the new polynomial, which has the claimed properties.

In this method as described, multiplication can violate our t-resilience because false behavior on the part of one of the agents can lead to erroneous values at many others through the communication of the

degree reduction. A further level of complication is required to ensure that the method described is in fact carried out correctly, but we will not go into that level of detail here (as before we refer the reader to [2, 1]). Intuitively, exchanges among the agents are used to prove that the agents follow the prescribed rules in conducting the transfer of information during the degree reduction phases. In order to simultaneously maintain t-resilience and t-privacy, we must be able to tolerate up to t incorrect shares at the beginning of a degree reduction (when the degree of the polynomials will be $2t$. So, we restrict t by $t < \lfloor \frac{m-1}{3} \rfloor$.

4 A scalable secret-bid second-price auction

We describe a second-price sealed-bid auction protocol which has the potential to be t-private and t-resilient, and which is efficient enough for use in large auctions. All polynomials used during the protocol are secret sharing polynomials from $Z_p[x]$ (the set of polynomials over the field of integers less than a prime p). The prime p should be chosen to support the assumption that the sum of several random non-zero elements from Z_p is very unlikely to be zero (i.e. that p^{-1} is very small). They are stored in a point-value representation, with each of the m auctioneers (who are indexed by i) holding the value at one point. Let $\alpha_i \in Z_p$ be the point held by the i^{th} auctioneer. These polynomials are manipulated by means of their evaluations at these m points.

Each of the n bidders (indexed by j) computes a set of secret sharing polynomials whose secrets encode her bid. The bidders distribute the shares of these polynomials among the auctioneers. The auctioneers then perform a multi-round computation to determine the selling price. The winning bidder is then determined by combining certain information from the auctioneers.

4.1 Bid submission

Each bidder selects a bid $b_j \in \{0, \ldots, V-1\}$ which is represented in some fixed base c as $b_j = b_{j1} \ldots b_{jd}$ (where $d = \lceil \log_c V \rceil$). This base c representation of the bid is then encoded with d ordered sets of $(c-1)$ secret-sharing polynomials, each of which has degree $t \leq \lfloor \frac{m-1}{3} \rfloor$. Each set of polynomials encodes one of the digits in a unary style as follows: Let $z \in \{0, \ldots, c-1\}$ be the value of the digit to be represented by a particular set. The first z polynomials are chosen such that their free coefficients are uniformly randomly selected from $\{1, \ldots, p-1\}$. The remaining $c - z$ polynomials are chosen with their free coefficients set to 0. We refer to the l^{th} polynomial of the set which encodes the k^{th} digit of the j^{th} bidder as s_{jkl}. We refer to the evaluation of this polynomial at the point controlled by the i^{th} auctioneer as $s_{jkl}(\alpha_i)$.

The bidders will use asymmetric keys to provide accountability and to enable the winner to claim her good. In the normal case, these keys will be the bidders' published public keys. (In the case where a bidder wishes to remain anonymous, a pseudonymous key can be used instead. Issues raised by anonymous bidders are addressed in section 4.6.4.)

Let M_{ij} be the string

$$s_{j11}, \ldots, s_{jkl}, \ldots, s_{jd(c-1)}$$

The bid submission messages are of the form

$$B_j \rightarrow A_i \ : \ E_{A_i}[M_{ij}], D_{B_j}[h(M_{ij})].$$

Note that we are ignoring lower level details such as message identifiers. D_q and E_q represent signature and encryption with an asymmetric key pair, and h is a crytographically secure hash function. The submission is verified to be a valid polynomial using the interactive protocol mentioned in section 3.

4.1.1 Incorrect bids

The selection method for the polynomials s_{jkl} requires their values at 0 to be either 0 or a uniformly random selection from $Z_p \setminus \{0\}$. This property is needed to ensure a low probability of error. Since p is prime, the property may be enforced by multiplying each s_{jkl} by an independent scalar value uniformly selected from $Z_p \setminus \{0\}$. Since this is multiplication by scalar values, the degree of the polynomials is not altered. Including this step does not eliminate the need for an honest bidder to select her polynomials randomly as described, since a non-random selection could leak information about the bids.

If $c > 2$, then the method of representing bids allows a bidder to submit a bid which does not correspond to a single value. This is due to the unary-style representation of the individual digits. We are unaware of any advantage that could be derived from using such malformed bids, but they could be easily prevented (if desired) as follows. After all bids are received, the auctioneers select a random number $r \in Z_p \setminus \{0\}$. The s_{jkl} should then be recomputed in order of descending l by

$$s_{jkl} \leftarrow s_{jkl} + r \cdot s_{jk(l+1)}.$$

For the base case, $l = c-1$, the polynomial $s_{jk(c-1)}$ remains unchanged. Since r is a scalar rather than polynomial, all these computations can be performed without any communication (other than agreeing upon r).

4.2 Bid resolution

The auctioneers use the submitted bids to compute the selling price b_0, which will be equal to the second highest bid submitted. This computation is performed over d rounds, one digit per round, going from most significant to least significant. We will use k to refer to the number of the current round. The value of the digits of b_0 is not kept secret (from the auctioneers), but is known to all auctioneers as it is computed.

Some shared polynomials u_{jk}, v_{jkl}, w_{jk} will be computed from the s_{jkl} over the course of price determination. Intuitively, these values are used to track the bid b_j of bidder B_j as compared to the first $k-1$ digits of the selling price.

- The value $u_{jk}(0)$ will be non-zero exactly when $\left\lfloor \frac{b_j}{c^{d+1-k}} \right\rfloor > \lfloor \frac{b_0}{c^{d+1-k}} \rfloor$.
- The value $v_{jkl}(0)$ will be non-zero exactly when $\left\lfloor \frac{b_j}{c^{d-k}} \right\rfloor \geq \lfloor \frac{b_0}{c^{d+1-k}} \rfloor c + l$.
- The value $w_{jk}(0)$ will be non-zero exactly when $\left\lfloor \frac{b_j}{c^{d+1-k}} \right\rfloor \geq \lfloor \frac{b_0}{c^{d+1-k}} \rfloor$.

While the comparisons above use the value of b_0, the results are determined entirely by the first $k-1$ digits, which are known prior to round k. The u_{jk} are indicators for the winning bidder. The value $u_{jk}(0)$ is non-zero when the first $k-1$ digits of the bid b_j indicate that the bid is greater than b_0. Since the selling price is the second highest bid, $u_{jk}(0) > 0$ means that bidder j has the highest bid. The w_{jk} are indicators for the losing bidders. The value $w_{jk}(0)$ will equal 0 when the first $k-1$ digits of the bid b_j indicate that the bid is less than b_0. So, $w_{jk}(0) = 0$ means that b_j is at most the third highest bid and that the particular bid b_j will have no effect on the outcome of the bidding.

The v_{jkl}, which are computed in round k by $v_{jkl} = u_{jk} + s_{jkl} \cdot w_{jk}$, indicate whether a bid is at least as high as some specific test value. The value $v_{jkl}(0)$ is non-zero when the first k digits of bid b_j (interpreted as a base c number) is greater than or equal to the first $k-1$ digits of b_0 appended by l (interpreted as a base c number). These v_{jkl} are used in round k to determine the k^{th} digit of the selling price by testing all possible values.

Since the u_{jk} and w_{jk} are computed from values $u_{j(k-1)}$, $v_{j(k-1)l}$, and $w_{j(k-1)}$ in round $k-1$, we must establish values for the u_{j1} and w_{j1}. For all j, let u_{j1} be the constant polynomial 0 and let w_{j1} be the constant polynomial 1. In an implementation, computation using these initial values will be trivial (i.e. addition to 0 or multiplication by 1).

The auctioneers will perform summations of the v_{jkl} over certain values of j (i.e. subsets of $\{1, \ldots, n\}$). Define $\{P_y : 1 \leq y \leq \lceil \log n \rceil\}$ to be a collection of subsets of $\{1, \ldots, n\}$ (the index set for the bidders). Let $j \in P_y$ if and only if the y^{th} bit (in a $\lceil \log n \rceil$ bit binary representation) of $j-1$ is zero. We will interpret these P_y as partitions which separate $\{1, \ldots, n\}$ into two parts based on inclusion in the set P_y. The rationale behind the selection of these partitions (the P_y for $1 \leq y \leq \lceil \log n \rceil$) is that for any pair of distinct bidders j and j', there is at least one partition which separates j from j'.

4.2.1 Phase 1: Determining the k^{th} digits of the bids

During the k^{th} round of resolution, the auctioneers compute the $n(c-1)$ values

$$\forall j, l . v_{jkl} = u_{jk} + s_{jkl} \cdot w_{jk}.$$

The products $s_{jkl} \cdot w_{jk}$ require a degree reduction step on all $n(c-1)$ products (these can be done simultaneously). This step is necessary only once one of the first $k-1$ digits of the bid is determined to be non-zero. Prior to this point, the value of $w_{jk}(0)$ is known to be 1 for all j, and the multiplication is unnecessary (in particular, it will never be necessary in the first round).

4.2.2 Phase 2: Determining the k^{th} digit of the selling price

Thew auctioneers now compute

$$\forall l . S_{kl} = \sum_{y=1}^{\lceil \log n \rceil} \left(\left(\sum_{j \in P_y} v_{jkl} \right) \left(\sum_{j' \notin P_y} v_{j'kl} \right) \right).$$

The inner summations may be performed without communication. Simple dynamic programming can be used to perform these summations in $3n(c-1)$ additions (of points in the field Z_p). The $(c-1)\lceil \log n \rceil$ multiplications require a degree reduction step on these $(c-1)\lceil \log n \rceil$ products. The outer summations may then be performed with $(c-1)(\lceil \log n \rceil - 1)$ field additions and no communication.

The outer summation runs over the indices for the partitions. The inner summations run over all j in

each half of a particular partition. These summations act like a logical OR operation over one side of the partition. In other words, if the partition side contains one or more bids which are at least as high as the test value, then the evaluation of this sum at 0 will be non-zero. For the polynomial result of multiplying the two inner summations, the evaluation at 0 will be non-zero only when the there is at least one bid on each side of the partition which is as high as the test bid. From this it is clear that no summand in the outer summation will have a value at 0 which is non-zero unless there are at least two bids which are at least as high as the test value. Conversely, we chose the partitions such that some partition would separate any pair j, j', so if there are two bids which are greater than the test bid, then at least one summand has a value at 0 which is non-zero. This means that (with high probability, see section 4.8) $S_{kl}(0)$ will be non-zero if and only if there are at least two bids at least as high as the test bid, or, equivalently, that the selling price is at least as high as the test bid.

By this stage, the auctioneers must have agreed on two further shared polynomials for each value of l: one of degree t (call it R_{kl}) which is selected uniformly randomly from $Z_p[x]$ and one of degree $2t$ (call it R'_{kl}) which is selected to be uniformly random up to the constraint that its value at 0 is 0. The k subscript is used to indicate that different polynomials should be chosen for each round. Since they are dependent only on m, c, and p, the R_{kl} and R'_{kl} may be generated in batches and used over the course of many auctions.

The auctioneers compute $\forall l.S'_{kl} = R'_{kl} + S_{kl} \cdot R_{kl}$. This computation has no influence on the determination of the winner or selling price. Rather, it is necessary to preserve secrecy of the bids when $S_{kl}(0)$ is revealed. The need for this step is not entirely theoretical; if it is not performed, there is an attack by a combination of an auctioneer and some bidders which could potentially reveal certain information about competing bids. Revealing S'_{kl} indicates only whether $S_{kl}(0)$ is zero or non-zero, which is exactly the information we will use to decide the k^{th} digit of the selling price.

Now, the secrets of these S'_{kl} are computed by the auctioneers (i.e., the values $\forall l.S'_{kl}(0)$). Let l_0 be the the largest value of l for which $S'_{kl} \neq 0$ ($l_0 = 0$ if $S_{kl} = 0$ for all l). The k^{th} digit of the bid is now set equal to l_0 (i.e. $b_{0k} \leftarrow l_0$).

4.2.3 Phase 3: Updating control values u and w

In preparation for the next round, the auctioneers assign

$$\forall j.u_{j(k+1)} = v_{jk(l_0+1)}, w_{j(k+1)} = v_{jkl_0}.$$

For the boundary conditions where v_{jkl} would be undefined: if $l_0 = 0$ then $w_{j(k+1)} = w_{jk}$; if $l_0 = c-1$ then $u_{j(k+1)} = u_{jk}$. This requires no communication and trivial computation.

4.3 Determining the winning bidder

Consider the case that the highest bid is strictly higher than the second highest bid. In this case, for exactly one value of j, $u_{j(d+1)}(0)$ will be non-zero. Some agent (possibly the auctioneers) can collect the $u_{j(d+1)}$ polynomials and determine the winner. Since there can be only one bidder with a higher price, this reveals no extra information.

If there is a tie for the highest bid, then $\forall j.u_{j(d+1)}(0) = 0$, and for at least two values of j, $w_{j(d+1)} > 0$. These values of j correspond to the tied bidders.

4.3.1 Breaking ties

Handling the special case where several biddders tie for the highest bid is surprisingly complex. First, simply detecting that there has been a tie is an information leak. This means that perfect privacy requires a method of computing the winner of a tied or untied bid in a uniform manner. Since computation and communication costs would leak information if they were different for tied or non-tied cases, this means that the overhead costs for all auctions must be the same as the overhead for the worst-case tie-breaking situation.

The simplest approach is to reveal $\forall j.u_{j(d+1)}(0)$ as described above to determine if there is a clear winner. If all these values are 0, then there is a tie. Now reveal $\forall j.w_{j(d+1)}(0)$ and randomly select some j for which $w_{j(d+1)} \neq 0$ as the winner. This method is direct and efficient, but reveals all tying bids (to the auctioneers).

Finding an efficient tie-breaking method to follow the bid resolution protocol as described is still an open problem. The authors are studying a modification to the overall protocol which allows efficient tie-breaking. The alternative has more complexity (in terms of its description) but uses a small value of p (with other modifications to prevent errors), so that communication costs are reduced.

4.4 A sample auction

We will give a simple example at a medium level of detail to illustrate the basic operation of the computation. We will not describe the lower level details of secure distributed computation. Let $c = 2$ (the radix), and $d = 3$ (the number of digits). This will allow for 8 bidding points, say $\{\$10, \$20, \ldots, \$80\}$. With base $c = 2$, one polynomial is sufficient to represent each digit of the bid. This means that $l = 1$ for all polynomials, so we will omit the l subscript. We will use $p = 7$, $m = 3$, $t = 1$, and $\forall i.\alpha_i = i$ for this example. Note that these parameters are chosen strictly for simplicity of the example; they are not secure.

Suppose three bidders B_1, B_2, and B_3 have bids $b_1 = \$20$, $b_2 = \$60$, and $b_3 = \$50$ respectively. In this case, the winner should be B_2 and the selling price should be the second highest bid, $b_3 = \$50$.

We use the symbols $\mathcal{T}$ and $\mathcal{F}$ to denote the values of polynomials' free coefficients. We will use the notation "$g = \mathcal{T}$" to mean that "$g(0) \neq 0 \pmod p$" and "$g = \mathcal{F}$" to mean that $g(0) = 0 \pmod p$.

During the bid submission step, each bidder selects the polynomials (s_j) to encode the digits of their bids. Each bid is encoded with one polynomial per digit (since $c = 2$). The bids are $b_1 \rightarrow s_{11}s_{12}s_{13} = \mathcal{FTF}$ and similarly $b_2 \rightarrow \mathcal{TTF}$ and $b_3 \rightarrow \mathcal{TFT}$.

The shared polynomials are distributed among the auctioneers, after which the auctioneers compute the result without further involvement from the bidders.

For example, the polynomials for b_1 could be $s_{11} = 4x + 0$, $s_{12} = 3x + 3$, and $s_{13} = 6x + 0$, and then bidder B_1 would share her polynomials by

- $B_1 \rightarrow A_1 \ : \ (4, 6, 6)$
- $B_1 \rightarrow A_2 \ : \ (1, 2, 5)$
- $B_1 \rightarrow A_3 \ : \ (5, 5, 4)$

Given the points of the shared polynomials, each auctioneer computes v_{jk} at k^{th} round by $v_{j(k+1)} = u_{jk} + s_{jk}w_{jk} \pmod p$. Table 1 shows how the polynomials are assigned through $k = 1, 2, 3$. The starting (i.e. $k = 1$) values are $\forall j.v_{j1} = s_{j1}, u_{j1} = \mathcal{F}, w_{j1} = \mathcal{T}$. As stated in the description, the values for the u_{j1} are w_{j1} just for consistency, they are known values and are only in the useless operations of multiplication by 1 ($\mathcal{T}$) or addition to 0 ($\mathcal{F}$).

- In round 1 ($k = 1$), there are at least 2 (exactly 2 in this case) j for which $v_{1k} = \mathcal{T}$, and thus $S_1 = \mathcal{T}$. This determines that the first digit of the selling price is 1 ($b_0 = 1\_$ or equivalently $b_0 \in \{4, 5, 6, 7\}$). Following the updating rule provided in section 4.2.3, auctioneers compute new polynomials $w_{j2} = v_{j1}, u_{j2} = u_{j1}$.

- In round 2, there is only one j for which $v_{2k} = \mathcal{T}$. So $S_2 = \mathcal{F}$, and the auctioneers update polynomials $\forall j.u_{j3} = v_{j2}, w_{j3} = w_{j2}$ accordingly. $S_2 = \mathcal{F}$ determines that the second digit of the selling price is 0, so $b_0 = 10\__2$ ($b_0 \in \{4, 5\}$).

- Finally, in round 3, there are two j for which $v_{3k} = \mathcal{T}$ so the auctioneers determine that $S_3 = \mathcal{T}$. We have $S_1 = \mathcal{T}$, $S_2 = \mathcal{F}$, and $S_3 = \mathcal{T}$, so $b_0 = 101_2 = 5$ and the selling price is \$50. Since $S_3 = \mathcal{T}$, we have $\forall j.u_{j4} = v_{j3}$. For exactly one value j_0 we have $u_{4j_0} = \mathcal{T}$. The winning bidder is B_{j_0}. By combining their shares of u_{jk} auctioneers determine that $j_0 = 2$; bidder B_2 is the winner.

4.5 Efficiency

In brief, the costs are non-trivial, and this protocol would not be appropriate for very low value (measured in cents) or extremely time-sensitive (results needed in seconds) auctions. However, most real-world auctions do not fall into these categories. Although the costs are not competitive with auctions which do not preserve privacy, such as [7], we believe that the complete privacy of the bids (and, if desired, the bidders)—even after the auction is completed—could well justify the higher implementation costs in many situations.

A major factor in the cost of the protocol is the fault tolerance. We describe the costs for $3t < m$, but if we were to accept slightly lower fault tolerance, the low-level implementation would be greatly simplified, resulting in a substantial savings in communication costs. For example, changing m from 4 to 5 (and making the improvements that this would allow) would reduce the volume of communication per auctioneer by nearly a factor of 10 in the examples in section 4.5.2.

Communication costs will be the dominant concern for the protocol described, and will consist largely of messages containing many points drawn from the field Z_p. To simplify the analysis, we will consider the costs in the case where there are no attempts to cheat; the cost to detect and correct cheating depends heavily on the number of attempted cheaters. Any cheating attempt (by at most t auctioneers) will be immediately detected and corrected, so cheating attempts by the auctioneers are unlikely to be frequent. Additionally, we will not

$k=1$	b_j	s_j			u_j	v_j	w_j
B_1	2	$\mathcal{F}$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{F})$	$\mathcal{F}$	$(\mathcal{T})$
B_2	6	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{F})$	$\mathcal{T}$	$(\mathcal{T})$
B_3	5	$\mathcal{T}$	$\mathcal{F}$	$\mathcal{T}$	$(\mathcal{F})$	$\mathcal{T}$	$(\mathcal{T})$
					$S_1 =$	$\mathcal{T}$	

$k=2$	b_j	s_j			u_j		v_j		w_j	
B_1	2	$\mathcal{F}$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{F})$	$(\mathcal{F})$	$\mathcal{F}$	$\mathcal{F}$	$(\mathcal{T})$	$\mathcal{F}$
B_2	6	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{F})$	$(\mathcal{F})$	$\mathcal{T}$	$\mathcal{T}$	$(\mathcal{T})$	$\mathcal{T}$
B_3	5	$\mathcal{T}$	$\mathcal{F}$	$\mathcal{T}$	$(\mathcal{F})$	$(\mathcal{F})$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{T})$	$\mathcal{T}$
						$S_2 =$		$\mathcal{F}$		

$k=3$	b_j	s_j			u_j			v_j			w_j		
B_1	2	$\mathcal{F}$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{F})$	$(\mathcal{F})$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$\mathcal{F}$	$(\mathcal{T})$	$\mathcal{F}$	$\mathcal{F}$
B_2	6	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{F}$	$(\mathcal{F})$	$(\mathcal{F})$	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{T}$	$\mathcal{T}$	$(\mathcal{T})$	$\mathcal{T}$	$\mathcal{T}$
B_3	5	$\mathcal{T}$	$\mathcal{F}$	$\mathcal{T}$	$(\mathcal{F})$	$(\mathcal{F})$	$\mathcal{F}$	$\mathcal{T}$	$\mathcal{F}$	$\mathcal{T}$	$(\mathcal{T})$	$\mathcal{T}$	$\mathcal{T}$
							$S_3 =$			$\mathcal{T}$			

Table 1: Computation of polynomials for example auction

consider some small messages (mostly involving optional counter-signatures on the bid submissions) which do not contribute substantially to the total bandwidth used.

The following efficiency analysis assumes that the protocol uses the absolute secret verification as described in [2, 1]. Each polynomial is distributed by sending two degree-t polynomials to each recipient. Each recipient then shares 1 point with each other recipient to verify the secret. As with much of the communication in the protocol, many points can be verified in parallel.

The multiplication and degree-reduction of a pair of polynomials can be compressesd into two rounds of communication (in addition to some preparation which may occur prior to the protocol and which is described below). In the first phase, each auctioneer verifiably distributes shares for a total of $t+3$ polynomials: polynomials encoding his shares of the two polynomials to be multiplied, a polynomial encoding his share of the result of the multiplication, and t other polynomials which enable confirmation that the product was correctly computed on that share. In the second phase, all the check points are exchanged among the auctioneers, ensuring that the secrets from the first phase are valid. Also, a different set of check values is broadcast by each auctioneer ($4t$ points each). These points ensure that a consistent (across all auctioneers) set of shares was used to compute the product polynomial. Given the honest majority, this ensures correct computation and enables correction of any errors (i.e. attempts to cheat). These two rounds combined require the communication of $2t^2+(m+7)t+3m+3$ for each ordered pair of auctioneers, and an additional $4t$ points of broadcast per auctioneer. The product of many pairs of polynomials may be computed in parallel.

The preparation for multiplication, as referred to above, is that the auctioneers must generate a random degree-t shared polynomial to be used for each multiplication. These polynomials can be generated in large batches prior to a run (or many runs) of the protocol. The process is simply for each auctioneer to generate and verifiably share one such polynomial, the sum of all m shared polynomials is one suitable random polynomial (even if t auctioneers are corrupt).

4.5.1 Costs for protocol phases

Given this verification model above, a bid submission consists of m messages (one to each auctioneer) from the bidder, each of size $2(t+1)d(c-1)$ points. The auctioneers then verify the submission by an all-to-all communication with messages of size $d(c-1)$ points.

Determining a digit of the selling price causes two occasions for degree reduction: the first when computing the v_{jkl} in phase 1, the second when computing the summands of S_{kl} in phase 2. This is a total of $(c-1)\cdot(n+\lceil\log n\rceil)$ multiplications over 4 communication rounds. There is one additional communication round where each auctioneer broadcasts $c-1$ points to reveal the S'_{kl} shares.

4.5.2 Example costs

Let us assume we have $m = 4$ auctioneers (so $t = 1$, no single auctioneer can cheat or violate privacy), and let $\log p \approx 96$, so each point is 12 bytes. We will estimate the communication costs (in terms of rounds, messages, and bytes) for two sample auctions. We will assume that all bids must be verified immediately (this maximizes the number of messages needed during bid submission), that the leading digit of the selling price is non-zero (worst-case). Costs will be counted as the total of sent and received for both bytes and messages.

Example 1: Consider the case with $n = 1000$ bidders, $V = 1024$ possible prices, and base $c = 2$ bid encoding (so $d = 10$).

- Bid submission. Each bidder sends 4 messages of 480 bytes each. Each auctioneer receives one of these messages and then sends 120 bytes to each other auctioneer. Total communication for each auctioneer during submission is 1.2MB in 7000 total messages.

- Bid resolution. In the first round of the protocol, phase 1 is unnecessary, so there will be 10 multiplications over 3 communication rounds. The remaining 9 protocol rounds each require 1010 multiplications and 5 communication rounds. Each multiplication requires 480 bytes of communication between each pair of auctioneers. Total communication for each auctioneer is $9100 * 960 * 3 = 26.2$MB in 288 messages over 48 communication rounds.

Example 2: Consider the case with $n = 50$ bidders, $V = 1024$ possible prices, and base $c = 4$ bid encoding (so $d = 5$).

- Bid submission. Each bidder sends 4 messages of 720 bytes each. Each auctioneer receives one of these messages and then sends 180 bytes to each other auctioneer. Total communication for each auctioneer during submission is $90K$ bytes in 350 total messages.

- Bid resolution. In the first round of the protocol, phase 1 is unnecessary, so there will be 18 multiplications over 3 communication rounds. The remaining 4 protocol rounds each require 168 multiplications and 5 communication rounds. Each multiplication requires 480 bytes of communication between each pair of auctioneers. Total communication for each auctioneer is $690 * 960 * 3 = 2.0$MB in 138 messages over 23 communication rounds.

4.5.3 Asymptotic efficiency

This protocol we describe uses $O(t^2 n \log V)$ communication for each auctioneer. Information theory tells us that the bids must be $O(\log V)$ in size. So, if we guarantee that t or fewer auctioneers can obtain no information about the bids, each bidder must distribute $O(t \log V)$ information to the auctioneers (which is $O(n \log V)$ per auctioneer). The verifiable secret sharing methods actually require an extra factor of $O(t)$ above the theoretical minimums for non-verifiable secret sharing. So, in total, just distributing the bids requires $O(tn \log V)$ communication for each auctioneer.

There are two major constant factors hidden by the notation which lie at the heart of high communication costs. The first is the large field we have chosen; the second is the constant associated with multiplying two polynomials. The choice of a large field was made to reduce the number of rounds of communicaiton required in phase 2, and alternative are being explored. The constant associated with multiplication is a more fundamental result of the distributed computation techniques used and the high fault-tolerance ($3t < m$) allowed.

4.6 Accountability

Given the application of these protocols to electronic commerce, we wish to ensure that the participants can be held accountable for their actions. By this, we mean that there are certain economic or legal consequences to the actions of the participants, and that there is sufficient non-repudiable evidence to prove what actions were taken and to enforce these consequences.

4.6.1 Deposits

In order to support non-repudiation, or at least to associate a cost with repudiation, bid submissions may include some electronic payment (e.g., digital cash) as a form of deposit. This is particularly useful in cases where there would be a difficulty holding the bidders accountable through a legal process (e.g., if the bidders are completely anonymous). The amount of the deposit may or may not depend on the value of the bid. If a bidder fails to accept the good, the deposit is forfeited and the auction is repeated. Depending on the particular payment system used for the deposit, some safe-guarding techniques may be used to prevent cheaters from redeeming deposits inappropriately. For example, verifiable signature sharing [6] may be used to require some threshold number of auctioneers to agree to redeem a digital

cash deposit. The use of verifiable signature sharing and digital payment deposits on bids first appears in [7].

There is a cost associated with such deposits beyond the computational cost of the implementation. The deposit is "in use", and is unavailable for other purposes during the course of the auction. The precise effect of having some amount in use is dependent on the digital payment scheme being used. In an anonymous digital cash scheme, for example, it is likely to mean that the funds have been withdrawn from a (possibly interest bearing) account. In any scheme, there is an opportunity cost to having the funds in use which must be taken into account. If the overall opportunity cost to the bidders is high (e.g. if the deposit is large or if there are many bidders), this will tend to discourage bidding and depress the selling price. Therefore, compensation for the auctioneer and cost to the bidders must be balanced when choosing the value of the deposit.

One implementation on deposits is for the seller to set fixed value for all bidders as the penalty for default. The appropriate magnitude for this default penalty is highly dependent on the specific characteristics of the seller, the good, and the bidders. This method is optimal in terms of the computation and communication costs and may be the best in practice.

Another method, not detailed here, would be to have the deposit match the value of the bid. This would introduce further computational complexity, particularly since the auction is second-price and so determining the value of the deposit, even for the winning bidder, would violate privacy. An efficient method for such deposits is an open problem (see section 5) Even discounting the computational overhead, this method yields optimal results only if the total of opportunity costs of all deposits is trivial.

4.6.2 Bidder accountability

The signature on the bid submission enables the auctioneers to prove the validity of a bid even if the majority of the auctioneers are not trusted. In the non-anonymous case, a well-known public key for the bidder should be used. In the anonymous case, the asymmetric key should be registered with some escrow agency or cryptographically linked to a deposit. Whatever key is used, the meaning of "bidder" in the following discussion is simply the entity which can generate signatures with the correct key. In all cases, the auctioneers must be able to prove that the bidder (by means of her asymmetric key) placed a bid at least as high as the selling price.

The simplest means of resolving a dispute is for the auctioneers to send all components to a trusted authority. The accuracy of the components can be verified by the signatures, and then the precise bid can be determined. The drawback of this method of resolution is that the authority will learn the exact value of the bid. However, for the auctioneers to inaccurately determine a winner, there must be a coalition of auctioneers large enough to determine the bid themselves, each member of which is corrupt or faulty. So the bid may already be known to the corrupt auctioneers.

Another method for resolving such a dispute is to perform a greatly simplified version of the auction. This simplified auction may be performed as a check by the auctioneers, to ensure that the winner was not selected incorrectly. Additionally, the shares possessed by the auctioneers (signed by the bidder) may be transferred to some other distributed authority, which will perform the check to resolve disputes. This simplified auction is performed with 3 bids: the disputed "winning" bid and two known bids b_0 and b_0-1 (where b_0 is the selling price determined by the disputed auction). If this auction returns a selling price of b_0, then the auction was correct; if it returns a selling price of $b_0 - 1$, it was incorrect. The result of the auction is always b_0 or $b_0 - 1$ and thus reveals no information other than whether the questioned bid is less than b_0.

4.6.3 Auctioneer accountability

The bid submissions can be signed (i.e. countersigned) by the receiving auctioneers and the signature made available to the submitting bidder. Then, if the selling price is published, any bidder who bid higher than the selling price can prove his superior bid. The same techniques described in section 4.6.2 for bidder accountability could be used to check that the bid is higher than the selling price without directly revealing the bid. In this case the two known bids for the simplified auction should be $b_0 + 1$ and b_0.

In this manner a bidder can prove a superior bid even if all auctioneers were corrupted (assuming that the bid was accepted in the first place). This method does not by itself protect against a denial-of-service attack by a large coalition ($> t$) of corrupt auctioneers unless they can be forced to accept and acknowledge a bid by some third party.

4.6.4 Anonymity

The protocol is designed to determine a selling price without revealing any of the bid values (other than

the second highest, of course). The identities of the bidding parties can also be protected. Anonymous (or pseudonymous) public keys may be used if there is some form of accountability to the keys which is considered acceptable for the purposes of the auction (possibly just a deposit as in 4.6.1). To preserve anonymity over real networks, the communications channel between the bidders and the auctioneers must be anonymized by means of some anonymous forwarding system.

4.7 Passive attacks

Due to the use of verifiable secret sharing, no coalition of at most t auctioneers can determine any information about the bidding from their shares of the bids. Similarly, the degree reduction steps preserve secrecy against coalitions of at most t auctioneers. But what about the polynomials S'_{kl} which are revealed in the course of determining the selling price? A single S'_{kl} is uniformly random and independent of all other variables except for its free coefficient. If $S_{kl}(0) = 0$, then $S'_{kl} = 0$. If $S_{kl} > 0$, then $S'_{kl}(0)$ is an element uniformly distributed over$Z_p \setminus \{0\}$. Note that $S_{kl}(0) > 0$ exactly when there are at least two bids whose value is at least the speculative selling price (the previously determined digits of b_0 together with k^{th} digit l).

4.8 Error analysis

During the course of computation, we occasionally assume that the sum of several uniformly random and independent non-zero values in Z_p is also non-zero. The probability that such a sum is 0 is at most $\frac{1}{p-1}$. Such errors may occur in the computation of the v_{jkl} or S_{kl}.

In the case of computing v_{jkl}, such a chance will introduce an error into the result of the bidding only if it occurs for the l that are relevant to the selection of b_0 (i.e., l_0 or $l_0 + 1$)). Also, $u_{jk}(0)$ will never be non-zero for any bidder other than the winner, so this error will be possible only for the winning bidder. The non-zero s_{jkl} are uniformly random (from $Z_p \setminus \{0\}$). They are also independent of $s_{jk'l}$ for $k' \neq k$, and thus are independent of the u_{jk} and w_{jk} (which are functions of the $s_{jk'l}$ with $k' < k$). Therefore, the value $s_{jkl} \cdot w_{jk}$ is uniformly random (from $Z_p \setminus \{0\}$). So the sum of u_{jk} and $s_{jkl} \cdot w_{jkl}$ (for any particular k and l) can equal zero with probability at most $\frac{1}{p-1}$.

Since the non-zero $s_{jkl}(0)$ are uniform and independent, the $v_{jkl}(0)$, if non-zero, will also be uniform and independent of any $v_{j'kl}(0)$ which is non-zero (for $j' \neq j$). In other words, the only dependency is contained in the property of being non-zero.

In the case of the S_{kl}, an error in the result of the bidding will be caused only for the one relevant l value l_0. The chance of incorrectly generating a S_{kl} with value 0 at 0 is at most $\frac{2}{p-1}$. To see this consider the values of S_{kl} as we compute it over increasing subsets of the bidders. Note the change in $S_{kl}(0)$ when we include the final bidder whose bid is as high as the test value (i.e. for whom $v_{jkl}(0) \neq 0$).

We can bound these sources of error by $\frac{2d}{p-1}$ from the v_{jkl} and $\frac{2d}{p-1}$ for the S_{jk}, totalling $\frac{6d}{p-1}$. We must choose p large enough that $\frac{4d}{p-1}$ is negligible. Since $d = \log_c V$, where V is the number of bidding points, its value in any implementation would be small (certainly less than 40). Making a tradeoff between speed and possibility of error, p should be in the range of 64-128 bits.

5 Open Problems

There are a variety of natural questions left unanswered by the work described. Below, we list a few natural directions for further work in this area. Work is under way on variations on the protocol which will address tie-breaking issues, reduce communication costs, and

1. *Tie-breaking*
   The protocol described does not provide efficient tie-breaking without some loss of privacy.

2. *Communication Costs*
   While we believe the communication costs are sufficiently small to make this protocol practical in many situations, low-value auctions are likely to play an increasingly important role in electronic commerce. Efficiency improvements that enable auctions with private bids in these low-value situations could be very useful.

3. *Hierarchical Auctions*
   In some situations, it might be advantageous to hold sub-auctions which partially determine the outcome of the auction. For example, each country might hold a sub-auction, with the leading candidates from each country participating in a final auction. In order to preserve privacy, the winners of the sub-auctions and their bids must not be revealed.

4. *Double Auctions and Auction Markets*
   A double auction is a more general form of auction where there are multiple sellers and multiple buyers. All parties tender bids and a market

clearing price is determined from those bids. A market clearing price is the equilibrium price at which the supply and demand (in units of the good) are equal. An auction market is a generalization of a double auction to continuous time. New bids are added and removed over time, causing the market clearing price to fluctuate. The stock market is a well known example of an auction market. Double auctions and auction markets are powerful market mechanisms, and privacy protecting protocols for these mechanism would be desirable. However, to be useful, such protocols must be highly efficient, particularly in the case of auction markets.

5. *Privacy vs. Performance*
The protocol described protects information privacy at the cost of greater overhead. It is possible that performance could be improved by relaxing some of the privacy constraints. What is the nature of this tradeoff? Depending on the context of the auction, relaxing privacy constaints could be appropriate.

References

[1] D. Beaver. Security, fault tolerance, and communication complexity in distributed systems. Technical Report HAR-CS-24-90, Computer Science Department, Harvard University, Boston, MA, 1992. Ph.D. thesis.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computing. In *Proceedings of the* 20^{th} *STOC*, pages 1–10. ACM, 1988.

[3] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. In *Proceedings of the* 20^{th} *STOC*, pages 11–19. ACM, 1988.

[4] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *STOC*, pages 383–395. ACM, 1985.

[5] David Clark. Internet cost allocation and pricing. In Lee McKnight and Joseph P. Bailey, editors, *Internet Economics*, Cambridge, MA, 1997. MIT Press.

[6] Matthew K. Franklin and Michael K. Reiter. Verifiable signature sharing. In L.C. Guillou and J. Quisquater, editors, *EUROCRYPT95*, pages 50–63. Springer-Verlag, 1995. Lecture Notes in Computer Science No. 921.

[7] Matthew K. Franklin and Michael K. Reiter. The design and implementation of a secure auction server. *IEEE Trans. on Software Engineering*, 22(5):302–312, 1996.

[8] O. Goldreich, S. Micali, and A. Wigderson. How to play and mental game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 218–229, May 1987.

[9] Andreu Mas-Colell, Michael D. Winston, and Jerry R. Green. Incentives and mechanism design. In *Microeconomic Theory*, New York, 1995. Oxford University Press.

[10] Lee McKnight and Joseph Bailey. *Internet Economics*. MIT Press, Cambridge, MA, 1997.

[11] Paul Milgrom. Auction theory. In *Advances in Economic Theory 1985: Fifth World Congress*. Cambridge University Press, 1985.

[12] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–614, November 1979.

[13] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16, 1961.

[14] Robert Wilson. Auction theory. In J. Eatwell, M. Milgate, and P. Newman, editors, *The New Palgrave*, London, 1987. MacMillan.