In Proceedings of of the First IEEE Workshop on Dependable and Real-Time E-Commerce Systems (DARE '98), June 1998, pp. 62-69

Multi-round Anonymous Auction Protocols

Hiroaki Kikuchi Michael Harkavy J. D. Tygar

Computer Science Department
Carnegie Mellon Univeristy
Pittsburgh, PA 15213
{kikn,bif,tygar}@cs.cmu.edu

Abstract

Auctions are a critical element of the electronic commerce infrastructure. But for real-time applications, auctions are a potential problem – they can cause significant time delays. Thus, for most real-time applications, sealed-bid auctions are recommended. But how do we handle tie-breaking in sealed-bid auctions? This paper analyzes the use of multi-round auctions where the winners from an auction round participate in a subsequent tie-breaking second auction round. We perform this analysis over the classical first-price sealed-bid auction that has been modified to provide privacy. We analyze the expected number of rounds and optimal values to minimize communication delays.

1 Introduction

Auctions are the most important market mechanism for setting prices. In an auction, a good can be sold at a price determined by interactions in the market. The Internet is a prime vehicle for supporting auctions. Moreover, auctions have been suggested as a basic pricing mechanism for setting prices for access to shared resources, including Internet bandwidth [CL97, MB97]. On the commercial side, there have been an increasing number of auctions held for consumer goods such as airplane tickets, and there are now a number of attempts to produce commercial auction software.

In addition to the real-time concerns associated with auctions, there are also privacy concerns. A corrupt auctioneer can derive detailed information about the bidders' preferences and the value they place on various goods. This is a serious drawback — consumers are naturally reluctant to give out personal information over the web, where they can not control who has access to the information or for what purposes it can be used. In particular, if an auctioneer can observe consumer behavior on an auction of a commodity good, he can often use shills to bid up a price arbitrarily.

Sealed bid auctions hold promise for real-time applications, since all bidders will submit their bids simultaneously. Thus, the time required for communication is limited. (Normally, one would term this as rounds of communication, but to avoid confusion with rounds of the auction, we will speak of it as phases of communication.)

Franklin and Reiter present a protocol for a sealed-bid auction[FR96]. Their protocol uses a set of distributed auctioneers and features an innovative primitive called verifiable secret-sharing. Their protocol successfully prevents a single auctioneer from altering a bid or throwing an auction to a single bidder. Unfortunately, their protocol also results in all auctioneers knowing all bids after the auction is decided. The natual question that arises is: can we hold a true private-bid auction?

Using a powerful set of theoretical computer science tools known as *secure distributed computation protocols* we can certainly answer this question affirmatively. (Some examples of secure distributed computation include Yao's millionaires protocol[YAO] which allows two parties to determine who is richer without revealing their wealth; Goldreich, Micali and Wigderson's protocols for bitwise AND and NOT using oblivious transfer[GMW87]; Chaum, Crepeau, and Damgard's protocol for computing XOR and AND based on the existence of *secure blobs*[CCD88]; Ben-Or, Goldwasser, and Wigderson's protocols for arithmetic operations $c \cdot x$, $x+y$ and $x \cdot y$ to simulate arbitrary logical circuits[BGW87]; and other protocols including [BFKR90, RB89, BMR90, BG89].) While these protocols can be used to simulate ar-

Hiroaki Kikuchi is currently at Tokai Univeristy, and was visiting faculty at CMU 1997-1998. The authors gratefully acknowledge support from 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.

bitrary circuits, and thus solve any computable problem, they require extensive communication and computation. They can have a dramatic explosion of communication phases — the number of communication phases can be a constant multiple of the depth of the circuit that performs the desired function. Improvements to the initial results have been made, but these methods are not immediately applicable to real-time auction applications. (The work in [HKT98] contains an effort to make these techniques usable in the auction setting).

In this paper, we consider an efficient protocol for electronic auctions based on a multiparty secret computation protocol. As with Franklin and Reiter's protocol, we use a distributed set of m auctioneers, so that any $m-1$ of them can not open a bid. (Note that in this paper, we are only dealing with passive attacks; that is groups of auctioneers, or eavesdroppers, who collaborate on information. For auction methods that deals with active attacks — that is auctioneers who might attempt to actively lie about the values they receive — see [HKT98, FR96].) However, in our protocol, the value of specific bids are kept secret even at the termination of the auction. Moreover, each round of the auction has a constant number of communication phases.

In each round of the auction, bidders can place a bid for a constant number of values k. For example, if we are bidding for an item, the first round of the auction may have $k = 10$ auction values of \$100, \$200, ... \$1,000. If the first round of the auction results in the maximum bid being a tie for a value of, say, \$400, then we place bids for a refined auction of \$400, \$410, ..., \$490. As we increase k, the size of each bid increases, but as we decrease k, we increase the likelihood of multiple rounds. To analyze the protocol for real-time auctions, we need to find optimal values of k.

2 Preliminary

2.1 Auction Styles

Auctions can be divided into different types:

- Public bids vs. secret bids

  In a *public bid auction*, all bids are known to other parties. For example, the classical *English auction*, the type one sees at Sotheby's or Christie's, each bidder announces his bid publically. Prices increase by a Δ increment.

  In a *secret bid auction*, such as a *sealed bid auction* the values of the bids are kept secret. Only the auctioneer knows the value of the bids.

  In this work, we go beyond the secret bid auction, to consider extremely secret auctions, where the value of the bid is held private even from the auctioneer.

- Constant time vs. time proportional to price

  A constant time auction requires a constant number of communication phases. For example, in a sealed-bid auction, we have one phase for bids to be submitted to the auctioneer, and one phase for the result to be announced.

  In contrast, many auction mechanisms, such as an English auction, or a Dutch auction can require many phases of communication. For example, in an English auction, the phases of communication can be proportional to the final price charged for the item.

  In this work, we aspire to find a single round auction. Unfortunately, if we have a tie, we require an additional auction round to break the tie. Thus we have a trade-off between the amount of information sent in each auction round and the probability that the auction will terminate, with no tie, at that round. This paper studies that trade-off under a variety of assumptions about the distribution of auction bids.

  Furthermore, please note that if we allow ourselves to have run-off rounds of the auction, we are no longer strictly adhearing to the sealed-bid max price auction. Instead, we are proposing something that is effectively a hybrid between traditional sealed-bid max price auctions and English auctions.

2.2 Requirements

We specify the following requirements for the auction:

Privacy No auction bid is revealed except for the winning bid.

Non-repudiation No winner can repudiate his bid. Otherwise, attackers could easily mount denial-of-service attacks.

(Note that this is weaker definition of non-repudiation than used by Franklin and Reiter; they use a deposit of digital cash to ensure that payment can actually be collected from the winner without his cooperation.[FR96].)

Efficiency We want the auction to run quickly.

3 Protocol Definition

3.1 Model

We assume n bidders, m auctioneers, and a seller. We assume that at most $t = m - 1$ auctioneers can conspire to try to reveal the value of a hidden bid.

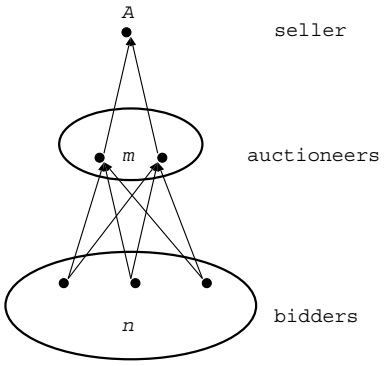

Figure 1: Auction model

3.2 Overview

The basic idea is based on secure addition [BGW87, RB89]. This protocol works as follows. A bidder prepares a bid-vector with a bid for each of k bidding prices. If his valuation is higher than a price, he bids his secret ID value; otherwise, he bids 0. The bid vectors are securely, privately added. There are three possible cases:

1. When only a single bidder bids at a particular price, the result equals the bidder's ID value. (Of course, only the bidder knows his identity.)

2. When more than one bidder bids at a particular price, the result is the sum of the bidders' ID values. The bidders can compare their bids with the sum vector, and see there are other competitors at the price.

3. When no one bids at a particular price, the result is 0.

Bidders' secret ID values are randomly generated for each price and are encrypted with the seller's public key in order to ensure anonymity. For example, the jth bidders ID value is

$$ID_j = E_A(D_A(j)||r)$$

where $D_A(j)$ is j's secret ID value digitally signed by an authority, A (the seller). $E_A(\cdot)$ is an encryption function with A's public key. The signed identity, $D_A(j)$, prevents a bidder from assuming a false identity. Independent random padding, r, is concatenated for each price, and k independent identities are generated, one for each price.

The entire bid vector containing k bids is signed by the bidder and sent in one phase of communication.

3.3 Protocol Definition

Protocol 1 Each auctioneer i will be associated with a distinct point $\alpha_i \in Z_p$ for the duration of the protocol.

Step 1: Polling. The seller publishes k prices, $\omega_1, \ldots, \omega_k$, for a good.

Step 2: Bidding The j-th bidder picks k random polynomials of the form

$$f_j^l(x) = s + a_1 x + \cdots + a_t x^t \pmod p$$

and sends $f_j^l(\alpha_i)$ to the i-th auctioneer ($j \in \{1, \ldots, n\}$, $i \in \{1, \ldots, m\}$, $l \in 1, \ldots, k$). The coefficients are uniformly randomly chosen for each polynomial. (Recall that t is the maximum number of conspiring auctioneers.) The free variable, s, is set to be ID_j if and only if he is willing to bid at price ω_l; otherwise, $s = 0$.

Step 3: Opening. The i-th auctioneer computes $F(\alpha_i) = f_1(\alpha_i) + \cdots + f_n(\alpha_i)$ for each of k prices, and sends the result to the other auctioneers and the seller. Given more than t points of the aggregate polynomial, $F(\alpha_1), \ldots, F(\alpha_m)$, each auctioneer uses Lagrange interpolation (in the style of Shamir secret sharing[Sh79]) or inverse-FFT (for certain choices of the α_i) to solve the simultaneous equations and obtain the free variable. This variable gives the sum of the identies of the bidders bidding at the given price.

Step 4: Declaring. The seller decrypts the winner's bid, ID_{j*}, with his private key, and retrieves winner's identity, j^*. After verifying the signature $D_A(j^*)$, the seller awards the item to the winner, j^*.

3.4 Example

We have three bidders, B_1, B_2 and B_3, and three auctioneers A_1, A_2 and A_3. The range of bidding value is $\{0, \ldots, 7\}$. Bidder B_1 bids 2 and picks 8 polynomials such that

$$f_1^0(0) = ID_1^0, f_1^1(0) = ID_1^1, f_1^2(0) = ID_1^2,$$
$$f_1^3(0) = \cdots = f_1^7(0) = 0 \pmod p$$

where f_j^2 denotes the j-th bidder's bid at price 2, and ID_j^2 is his 2nd secret ID (recall that each secret ID had a unique random padding.) Suppose that bidder B_2 and B_3 bids 6 and 5, respectively. After distribution of bids, auctioneer A_1 adds three polynomials for each of 8 prices, and publishes the result, which is a point of the aggregate polynomial F given by;

$$F^l(\alpha_1) = f_1^l(\alpha_1) + f_2^l(\alpha_1) + f_3^l(\alpha_1) \pmod p.$$

for each $l \in \{0, \ldots, 7\}$. In the same way, all auctioneers publish the 3 different points of the polynomial F, and have the result as follows;

$$\begin{aligned}
F^0(0) &= ID_1^0 + ID_2^0 + ID_3^0 \pmod p \\
F^1(0) &= ID_1^1 + ID_2^1 + ID_3^1 \pmod p \\
F^2(0) &= ID_1^2 + ID_2^2 + ID_3^2 \pmod p \\
F^3(0) &= ID_2^3 + ID_3^3 \pmod p \\
F^4(0) &= ID_2^4 + ID_3^4 \pmod p \\
F^5(0) &= ID_2^5 + ID_3^5 \pmod p \\
F^6(0) &= ID_2^6 \pmod p \\
F^7(0) &= 0 \pmod p
\end{aligned}$$

This case shows the highest bid is 6 and the winner is the second bidder. Note that every auctioneer will know only the highest bid. Only the seller, and the winner, will know the identity of the winner.

3.5 Simplified Protocol

Instead of using a secret-sharing scheme, we can use $m-1$-wise independent values to compute the sum of the vector-bids.

Protocol 2

Step 2: Bidding. The j-th bidder chooses a $m \times k$ random matrix B_j

$$B_j = \begin{pmatrix} b_1^j(1) & \cdots & b_1^j(i) & \cdots & b_1^j(m) \\ \vdots & & \vdots & & \vdots \\ b_k^j(1) & \cdots & b_k^j(i) & \cdots & b_k^j(m) \end{pmatrix}$$

where each row $(l = 1, \ldots, k)$ satisfies

$$\sum_{i=1}^{m} b_l^j(i) = \begin{cases} 0 \pmod p & \text{if } v_j < \omega_l\ , \\ ID_j^l \pmod p & \text{if } v_j \geq \omega_l\ , \end{cases}$$

and where v_j is j-th bidders valuation. The j-th bidder sends the i-th auctioneer this vector: $b_1^j(i), \ldots, b_k^j(i)$. After receiving the bids, each auctioneer will sum all bids he has recieved, calling the summed vector $c_1(i), \ldots, c_k(i)(i = 1, \ldots, m)$. All bidders commit to the vector using a one-way function and publish the results. The sum for ω_l is defined by

$$c_l = c_l(1) + \cdots + c_l(m) \pmod p$$

where $c_l(i) = b_l^1(i) + \cdots + b_l^n(i) \pmod p$.

Step 4: Declaring. Let c_{j^*} be the highest sum. For the price ω_{j^*}, if there exists a single bidder j at the price, then the sum c_{j^*} is equal to his secret ID, namely $ID_{j^*}^i$. The winner can verify this by checking whether $c_{j^*} = ID_{j^*}^i$. The seller decrypts the sum c_{j^*} and check if its validity; if there was more than one bid at the given price, then with high probability, the value will not decrypt properly.

3.6 Secret and Multiple-rounds Auction

We can simulate a number of different auctions with this technique. The first three of these are strawman protocols — only the fourth, a generalized tree structure, would generally be practical.

1. Secret English auction
   We hold one auction round for each bidding price; if there is a tie, we continue to the next bidding price. This gives us a nice, slow, auction that acts like an English auction, but preserves privacy.

2. Secret Dutch auction
   Similarly, we can have descending prices, until one bidder places a bid. Again, this method is slow.

3. Binary tree auction
   Set V be the highest valid bidding value. Consider a set of biding domain, $\{\omega_1, \ldots, \omega_V\}$, into two interval $\{\omega_1, \ldots, \omega_{V/2}\}$, and $\{\omega_{V/2+1}, \ldots, \omega_V\}$. Each party bids on the two sets, and if the the higher interval contains more than one bid, recurse on the more restricted domain $\{\omega_{V/2+1}, \ldots, \omega_V\}$; otherwise, recurse on the other (lower-value) interval. The auction ends when exactly one bidder is left on the higher interval.

4. Hierarchical auction
   Generalize the "binary tree auction" by replacing the binary price with k polling prices defined by $V/k, 2V/k, \ldots, (k-1)V/k$. The ith slot indicates a bid in the range. $[(i-1)V/k, iV/k)$. Look at the highest region receiving a bid. If it has one bid, we have found the winning bid; otherwise if it has more than one, recurse by dividing the winning range into a further k sub-divisions. Figure 2 shows an example of a 3-tree auction.

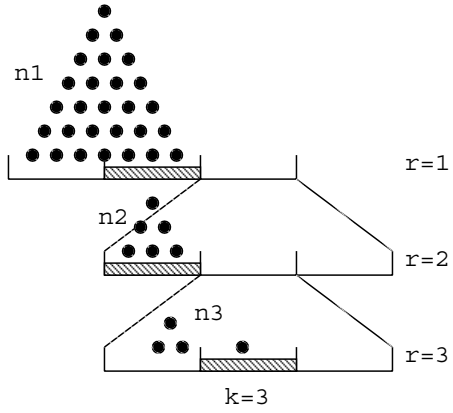

Figure 2: Processing in a k-tree auction

As k increases, the number of rounds decreases, but the length of the bid vector increases.

> Question: what value of k is optimizes the expected cost of the entire auction? (Note that there is no reason that k must be the same across different rounds; there may be situation in which varying k between rounds would be most efficient, but they do not appear in the relatively simple model we are using).

4 Estimation

4.1 Renormalizing Arbitrary Distributions

As we mention above, this auction protocol has a tradeoff. We can allow more fine-grained bids, but this will (linearly) increase the length of each bid sent in each round of the auction. We can switch to more coarse-grained bids, but this can result in the likelihood of more auction rounds. What is the optimal tradeoff?

The answer, of course, depends heavily on the probability distribution of the bids. If we know the probability distribution, we can figure out the optimal distribution of bids. (Throughout this discussion, we are assuming that bids are independent.)

Now, some parties might object. Isn't the need for an auction based on the uncertainty in the bids. If we knew this in advance, we don't need an auction; we can simply set a fixed price.

The truth, we believe, lies somewhere in the middle. In the case of a commodity (such as RSVP'd network bandwidth) that is repeatedly auctioned, it will usually be the case that the probability distribution on the bids will move slowly between bids. If this is the case, then we can set an appropriate probability distribution, and revise it as necessary. Each auction round will consist of a series of ranges $[0, \omega_1), [\omega_1, \omega_2), \ldots, [\omega_{k_2}, \omega_{k_1}), [\omega_{k-1}, \infty)$. (This raises a bit of a paradox — giving the participants the expected distribution leaks information from the seller to the buyers. However, if we anticipate that all this information is derived from successive private auctions, then we don't have any leakage of information from sellers to buyers.) Now, if our probability distribution on a bid is $g(x)$, then we want to set the $\omega_1, \ldots, \omega_k$ so that

$$\int_{\omega_i}^{\omega_{i+1}} g(x)\, dx = 1/k.$$

This, in effect, renormalizes $g(\cdot)$ so that it acts like a uniform distribution. So, we now consider the case where bids are independently, uniformly distributed.

4.2 Number of Tied Winners

Let n be the number of bidders and k be the dividing factor. If the bids are uniformly, independently distributed, then the probability that a particular slot has a highest bid for a particular bidder is $1/k$. Given n and k, the probability that u bidders have the same highest bid is given by $P_{n,k}(u) =$

$$\sum_{i=0}^{k-1} \left(\frac{k-i}{k}\right)^n \binom{n}{u} \left(\frac{1}{k-i}\right)^u \left(1 - \frac{1}{k-i}\right)^{n-u}$$

Figure 3 shows the probability density function of $P_{n,k}(u)$ where $n = 12, k = 3$. The density function reaches its maximum at 4, which is approximated by $L[T] = n/k = 12/3 = 4$.

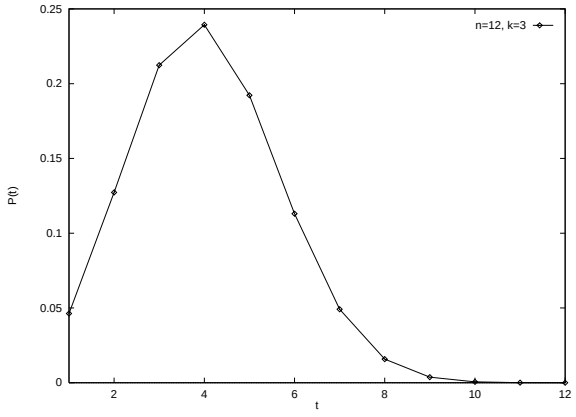

Figure 3: Density function: number of tied winners

The auction completes when the highest slot has just one bid, or $u = 1$. Setting $u = 1$, we have a probability

of completion given by

$$P_{success}(n,k) = \sum_{i=0}^{k-1} \left(\frac{k-i}{k}\right)^n \frac{n}{k-i} \left(1 - \frac{1}{k-i}\right)^{n-1}$$

We illustrate the the probability where $k = 5$ in Figure 4

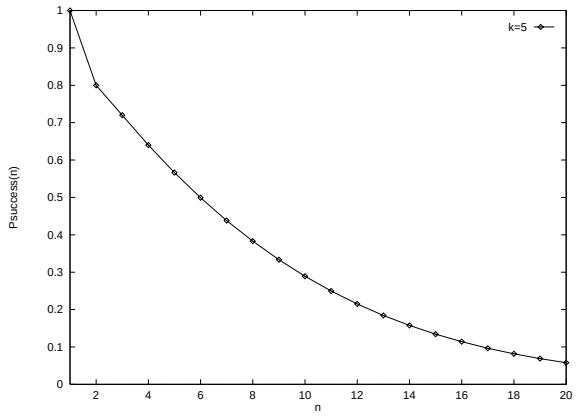

Figure 4: Probability of completion

4.3 Approximation of Number of Rounds

What is the expected number of rounds given n and k?

Summing over the above equation for $P_{n,k}(u)$, we can approximate as follows:

$$P_{n,k}(u) \simeq \left(\begin{array}{c} n \\ u \end{array} \right) \left(\frac{1}{k}\right)^u \left(1 - \frac{1}{k}\right)^{n-u}$$

But note that this is just the binomial distribution with mean of n/k, and variance $n(k-1)/k^2$. The expected number of tied winners is $1 + n/k$, and this, in turn is the expected number of bidders of the next round. As a first approximation, we have the expected number of winners after i rounds as

$$n_i = n_{i-1}/k = n_0/k^i$$

where $n_0 = n$, the original number of bidders in the first round. The auction ends when just one winner remains, hence, the expected number of rounds is $\log_k(n)$.

4.4 Expected Number of Rounds

The above analysis used an approximation. Suppose we did not use that approximation? At round i, the expected number of tied bidders will be $\sum_{u=1}^{n_i} P_{n_i,k}(u)u$, and this, in turn, is the initial population n_{i+1} for the next round $i+1$. The value will decrease exponentially with the number of rounds (see the example where $n = 100$ and $k = 2$ in

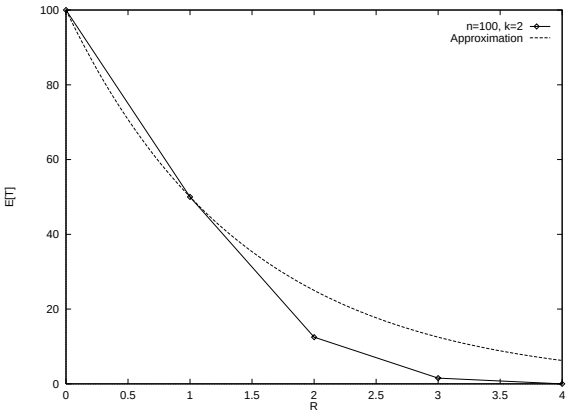

Figure 5: Expected number of tied winners

Figure 5 — the true value is given by the solid line; the approximation by the dashed line.)

Similarly, the expected number of rounds given n and k is

$$\sum_{r=0}^{\infty} P_{success}(E[N_{r-1}], k)r$$

This is shown in Figure 6 for $n = 10$ (again, the approximation is shown with a dashed line.)

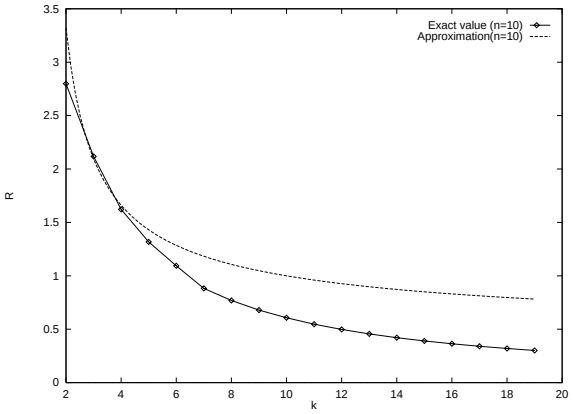

Figure 6: Expected numbers of rounds

4.5 Communication Costs

What is the optimum dividing factor k? Let C_A and C_B be the bandwidth of the auctioneer and the bidders respectively. (Typically C_B will be greater than or equal to C_A.) Using the polynomial method discussed in section 3.3, a bidder sends $Qm(k-1)$ bits per round in a channel of C_B bps, where Q is the number of bits in the modulus. Note we can slightly reduce the message length, by using

$(k-1)$ instead of k, because submitting a vector of zero bids can be thought of as bidding the lowest price. On the other hand, an auctioneer receives $Qn(k-1)$ bits each round in a channel of C_A bps. We must also take into account the period of time required to begin the round and begin receiving bids. Compared to the cost of receiving a bit, this takes enormous amount of time, say L sec. As k increases, the expected number of rounds decreases and the cost per round increases. We use the result of the approximation above to estimate an expected value of rounds. Eventually, the total time to complete the auction taken by an auctioneer, $T_A(k)$, is given by,

$$T_A(k) = (L + \frac{Qn}{C_A}(k-1)) \log_k(n)$$

and the total time spend by a bidder, $C_B(k)$, is

$$T_B(k) = (L + \frac{Qm}{C_B}(k-1)) \log_k(n).$$

Differentiating by k, we have

$$\frac{d}{dk}C_A(k) = \frac{(1-k)nQ - LC_A}{k(\log(k))^2} + \frac{nQ\log(n)}{\log(k)} = 0,$$

which can be simplified as

$$\frac{LC_A}{Qn} = k\log k \log n - k + 1$$

The dividing factor k satisfying this equation minimizes the total expected time to complete the auction.

In Figure 7, we show a particular behavior of $T_A(k)$ and $T_B(k)$ provided with the following parameters;

L	$=$	$240s$	Q	$=$	100bits
C_A	$=$	10Mbps	n	$=$	10^5
C_B	$=$	28.8kbps	m	$=$	10

With these constants, the time delay of an auctioneer is greater than that of a bidder, so the bottleneck is at the auctioneers. By applying the above equation, we see that $T_A(k)$ is minimized when

$$k^* = 72.72.$$

which means that the expected time for the complete auction is about 14 minutes.

5 Conclusion

This is only a partial attempt to address the intersecting questions of computational performance, economic efficiency, and privacy in electronic auctions. Given the importance of the area, more work is needed. Our approach in this paper was a narrow one; we discussed a particular approach to electronic auctions and attempted to make reasonable performance estimates for our model. The applicability of auctions to real-time systems remains to be demonstrated, but we hope that our study sheds light on some of the tradeoffs of particular tie-breaking strategies.

In a separate research paper [HKT98], we investigate electronic auctions with strong security and economic properties. In that paper, we study second price (Vickrey) auctions that have strong economic efficiency properties, while allowing fully private bids, so that auctioneers and participants can gain no information about the distribution of the bids. In that paper, we also address the question of cheating by participants who may try to subvert the auction.

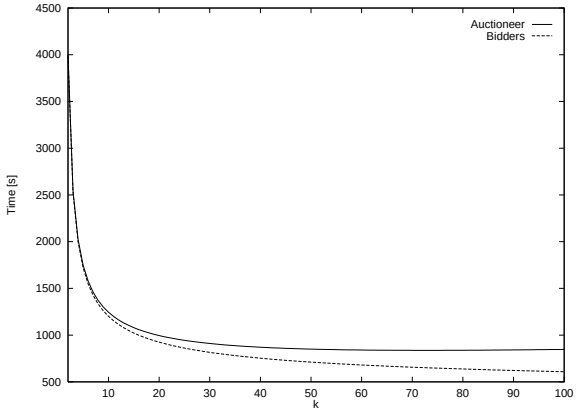

Figure 7: Time to run auction, in the example in Section 4.4

References

[MW] Andreu Mas-Colell, Michael D. Whinston and Jerry R. Green, *Microeconomic Theory*, Oxford university press, 1995, pp. 857-925

[FR96] Matthew K. Franklin and Michael K. Reiter, The Design and Implementation of a Secure Auction Service, *IEEE Transactions on Software Engineering*, 22(5), pp. 302-312, 1996.

[BFKR90] D. Beaver, J. Feigenbaum, J. Kilian and P. Rogaway, Security with Low communication overhead, Crytpo'90, 1990, pp.62-76

[BG89] D. Beaver and S. Goldwasser, Multiparty Computation with Faulty Majority, Proc. of FOCS, 1989, pp.468-473.

[BMR90] Donald Beaver, Silvio Michali, and Philip Rogaway, The round complexity of secure protocols, STOC, 1990, pp.503-513

[BGW87] M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, STOC88, pp.1-10

[BF97] Dan Boneh and Matthew Franklin, "Efficient Generation of Shared RSA Keys," *Advances in Cryptology -CRYPTO'97*, Springer-Verlag, 1997, pp. 425-439

[Ch88] D. Chaum, "The Dining Cryptographer Problem: Unconditional Sender and Receiver Untraceability," *Journal of Cryptology*, v.1, n.1, 1988, pp.65-75

[CCD88] D. Chaum, C. Crepeau and I. Damgard, Multiparty unconditionally secure protocols, STOC88, 1988, pp.11-19

[CDG87] D. Chaum, I. Damgard, J. van de Graaf, Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result, CRYPTO'87, LNCS 298, 1987, pp.87-119

[CL97] David Clark, Internet Cost Allocation and Pricing, *Internet Economics*, MIT Press, 1997

[EGL85] S. Even, O. Goldreich, and A. Lempel, A Randomized Protocol for Signing Contracts, *CACM*, 28, 6, pp.637-647, 1985

[FS87] A. Fiat and A. Shamir, How to Prove Yourself: Practical Solutions to Identification and Signature Problems, Proc. of Crypto'86, LNCS 263, 1986, pp.186-194

[GMW87] Oded Goldreich, Silvio Micali and Avi Wigderson, How To Play Any Mental Game or A Completeness Theorem for Protocols with Honest Majority, *ACM STOC*, p.218-229, 1987

[HKT98] Michael Harkavy, Hiroaki Kikuch, and J. D. Tygar, Electronic Auctions with Private Bids. To appear, Proceedings of the 3rd USENIX Workshop on Electronic Commerce, August 1998

[MM] R. Perston McAfee and John McMillan, "Auctions and Bidding," *Journal of Economic Literature*, 1987, 25, pp.699-738

[MB97] Lee McKnight and Joseph Bailey, *Internet Economics*, MIT Press, 1997

[Mi] Paul Milgrom, "Auctions and Bidding: A Primer," *Journal of Economic Perspectives*, vol.3, No.3, pp.3-22, 1989

[RB89] T. Rabin and M. Ben-Or, Verifiable Secret Sharing and Multiparty Protocols with Honest majority, STOC'89, 1989, pp.73-85

[Sh79] A. Shamir, How to share a secret, CACM, 22, 1979, pp.612-613

[YAO] Yao, A.C., Protocols for secure computations, In Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, 1986, pp.162-167