In Institute of Electronics, Information, and Communication Engineers Transactions on Information and Systems, E82-D:4, April 1999, pp. 769-777

PAPER *Special Issue on Internet Technology and Its Applications*

Multi-Round Anonymous Auction Protocols

Hiroaki KIKUCHI[†], ***Member***, **Michael HAKAVY**[††], ***and*** **Doug TYGAR**[†††], ***Nonmembers***

SUMMARY 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 full anonymity. We analyze the expected number of rounds and optimal values to minimize communication costs.

key words: *auction, multiparty computation, anonymity, communication cost*

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 [8], [14]. 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. Bidders will bid up to their *indifference price* — that is, the price at which they value the good being auctioned. A corrupt auctioneer can thus 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.

Franklin and Reiter present a protocol for a sealed-bid auction [11]. Their protocol uses a set of distributed auctioneers and features an innovative primitive called verifiable signature-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 the full bid of all the bidders at the end of the auction. The natual question that arises is: can we hold a truly anonymous auction?

Sealed bid auctions hold a great 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, I will speak of it as phases of communication.)

Using a powerful set of theoretical computer science tools known as *secure function computation protocols* we can certainly answer this question affirmatively. (Some examples of secure function computation include Yao's millionaires protocol [20] 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 [13]; Chaum, Crepeau, and Damgard's protocol for computing XOR and AND based on the existence of *secure blobs* [6]; Ben-Or, Goldwasser, and Wigderson's protocols for arithmetic operations $c \cdot x, x + y$ and $x \cdot y$ to simulate arbitrary logical circuits [3]; and other protocols including [2], [4], [5], [18].) While these protocols can be used to simulate arbitrary circuits, and thus solve any computable problem, they require extensive communication and computation. They usually have a dramatic explosion of communication phases — the number of communication phases is at least a constant multiple of the depth of the circuit that performs the desired function. Clearly, this work, while seminal, is not immediately applicable to real-time auction applications.

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. However, in our pro-

Manuscript received September 29, 1998.
Manuscript revised January 5, 1999.

[†]The author is with the Faculty Member in the Department Electrical Engineering, Tokai Univeristy, Hiratsukashi, 259–1292 Japan.
[††]The author is Ph.D. student in the School of Computer Science, Carnegie Mellon Univeristy, Pittsburgh, PA 15213–3891, U.S.A.
[†††]The author is a Professor of Computer Science and Information Management at University of California, Berkeley, CA 94720–4600, U.S.A.

tocol, 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 $\$100, \$200, \$300, \$400, \ldots, \$1,000$. 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 $\$400, \$410, \$420, \ldots, \$490$. 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.

In this paper, we review some styles of auction and gives fundamental requirements for electronic auction in Sect. 2. After we describe secure multiparty computation, the primitive protocol for first-price auction is defined and demonstrated with a simple example. We also present a simplified version of auction protocol based on pairwise independent random variables. Based on these protocol, we propose a protocol for multi-round anonymous auction in Sect. 3. Section 4 gives probabilistic properties of our proposed protocol and an expected number of rounds given k and n. These estimate could be useful for designing a practical auction system which takes into account not only security but also efficiency in terms of communication overhead.

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 auctions, we have one round for bids to be submitted to the auctioneer, and one round for the result to be announced.
  In contrast, many auction mechanisms, such as an English auction, or a Dutch auction can require multiple 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 bids. In fact, we believe that these assumptions may be unrealistic, but the style of analysis we propose could illuminate techniques for finding optimal strategies for conducting auctions under different distributions.
  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 winning (this includes the case of bids that win a single round of the auction and tie.)

Anonymity No one (except the seller) can know the winner.

Non-repudiation No winner can repudiate his bid. (Otherwise, we could imagine a bidder who placed multiple bids, and cancelled all but the minimum required to secure the good.)
(Note that this is weaker definition of non-repudiation than used Franklin and Reiter; they use a deposit of digital cash to ensure that payment can actually be collected from the winner without his cooperation [11].)

Validity No one can submit invalid bid which affects the outcome of the auction without being detected. Note that detection is necessary but not sufficient for avoiding misbehavior of bidders, because there are protocols to correct any misbehavior made by at most t bidder's conspiracy [3].

Efficiency We want the auction to run fast. To measure efficiency, we use *round complexity* to denote the expected number of rounds in execution of a protocol and *communication complexity* is a total messages in bits sent among bidders during an execution [3].

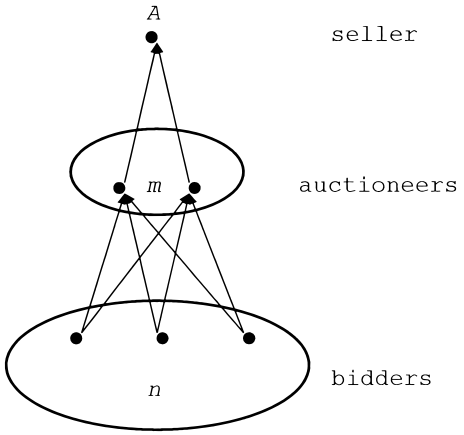

Fig. 1 Auction model.

3. Protocol Definition

3.1 Model

Suppose that we have n bidders, m auctioneers, and a seller. We assume that at most $t-1$ auctioneers may be faulty.

3.2 Overview

The basic idea is based on secure multiparty computation of addition [3], [18].

Intuitively, this protocol works as follows. A bidder prepares multiple bids for each of k bidding prices. If his valuation is higher than a price, he bids his secret ID value; otherwise, he bids 0. With the sequence of bids as input, the bidders participate in the secure multiparty computation for addition, which figures out the sum of some identities of bidders who are willing to bid at the price. The following three cases happen:

1. When a single bidder is willing to bid at a price, the result would equal to the winner's identity. No one except the winner can know whose identity it is. Therefore, the anonymity is established in this protocol.
2. When more than one bidders are willing to bid at a price, the result is the total of the bidders identities. By comparing the result with each bid, bidders can know there is other competitors at the price.
3. When no one bids at a price, the result is 0. This always happen at impossibly higher price. No information is leaked.

Bidder's secret identities are assigned for each prices and somehow encrypted with the seller's public key in order to ensure the anonymity. For example, jth bidder's identity can be defined by

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

where $D_A(j)$ is the j's secret identity digitally signed by an authority, A, (the seller) as a proof of authorized bidder, and $E_A(\cdot)$ is an encryption function with A's public key. The signed identity, $D_A(j)$, is necessary to prevent cheating with anyone else's identity to bid as his identity. Concatenated with random padding, r, for each price, k independent identities are generated. If the winner's anonymity against the seller is not necessary, any symmetric key encryption or PIN can be used instead of digital signature.

In terms of round complexity, all k bids can be sent in batch, thus one round is involved in this protocol.

3.3 Protocol Definition

Protocol 1

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

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

and sends $f_j(\alpha_i)$ to i-th auctioneers ($j \in \{1, \ldots, n\}, i \in \{1, \ldots, m\}$). The degree t is a maximum number of faulty auctioneers to be considered. The free variable, s, is set to be ID_j if and only if he is willing to bid at a price; otherwise, $s = 0$.

Step 3: Opening. i-th auctioneer computes $F(\alpha_i) = f_1(\alpha_i) + \cdots + f_n(\alpha_i)$ for each of k prices, and publishes the result to other auctioneers and the seller. Given more than t points of the aggregated polynomial, $F(\alpha_1), \ldots, F(\alpha_m)$, each auctioneer uses LaGrange scheme to solve the simultaneous equations and obtains the free variable, which gives the sum of identities of bidders who are willing to bid.

Step 4: Declearing. 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 identity with different random padding to others. 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 aggregated polynomial F given by;

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

for each $k \in \{0, \ldots, 7\}$. In the same way, all auctioneers publishes 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 can know the the highest bid, while no one knowwho is the winner except the seller and the winner himself.

3.5 Simplified Protocol

Instead of polynomial, we can use $m-1$-wise independent value to compute the total of each secret.

Protocol 2

Step 2: Bidding. The j-th bidder choose an $m \times k$ random matrixes 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

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

where v_j is j-th bidders valuation. The j-th bidder sends i-th auctioneer $b_1^j(i), \ldots, b_k^j(i)$. (Note that $b_1^j(j), \ldots, b_k^j(j)$ are not sent to other bidders.) After exchange bids, each auctioneer sums all bids he has recieved for each price and call them $c_1(i), \ldots, c_k(i)(i = 1, \ldots, m)$. All bidders commit to their bids by publishing the results of a cryptographic hash. 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 willingness $ID_{j^*}^i$. The winner can know it by checking $c_{j^*} = ID_{j^*}^i$. The seller decrypts the sum c_{j^*} and checks if it is valid or not; if some bidders are tied with the highest price $j*$, the c_{j^*} is sum of encryptions, which spoils the decryption.

3.6 Secret and Multiple-Rounds Auction

1. Secret English auction
   The protocol is run for each bidding price, which is raised until only one bidder bids to the price. The difference to the standard English auction is the result known to bidders is either of the followings:
   a. noone bids to the bidding price.
   b. there are some (at least one) bids at the price.

   In the latter case, all bidders have to hold on the next round until there is a single bidder bidding to the price. The bidders who bid the current price can know whether he is tied with someone else or not, but he can not know who it is.
2. Secret Dutch auction
   As the same way to the English auction, the bidding price is going down until some bidder bids the value. It is almost same to the standard Dutch auction, except that the winner can be made anonymous.
3. Binary tree auction
   Let V be the highest valid bidding value. Split a set of biding domain, $\{\omega_1, \ldots, \omega_V\}$, into two interval $\{\omega_1, \ldots, \omega_{V/2}\}$, and $\{\omega_{V/2+1}, \ldots, \omega_V\}$ and use Protocol 1 or 2 with just one price, $\omega_{V/2}$. If the higher interval contains more than one bid, repeat Protocol with restricted domain $\{\omega_{V/2+1}, \ldots, \omega_V\}$; otherwise, examine the other interval. The auction ends if a single bidder is left at the higher interval.
4. Hierarchical auction
   Generalize "binary tree auction" with k polling prices defined by $V/k, 2V/k, \ldots, (k-1)V/k$. We call ith slot to mean $[(i-1)V/k, iV/k)$. The protocol is executed multiple times for each of k slots in one round. In the next round, the set of prices is restricted to the previous higest slot for which some bids were submitted, and this range is divided into k subslots. Figure 2 shows an example processing the hierarchical auction with $k = 3$.

The dividing factor, k, in the hierarchical auction influences the round and the communication casts. As k increases, the number of round to complete the protocol is going shrinks. Extremely, maximizing k results in the least round complexity, but the bandwidth spent by the protocol grows.

Question: what value of k is likely to optimize the bit complexity involved the whole auction?

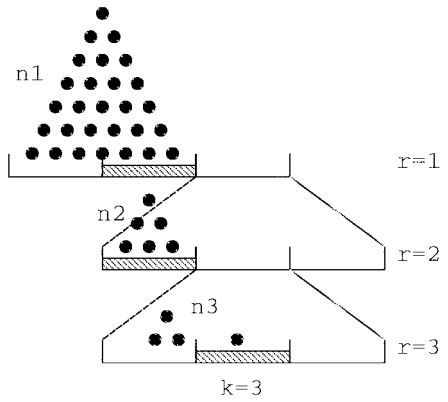

Fig. 2 The processing in the hierarchical auciton.

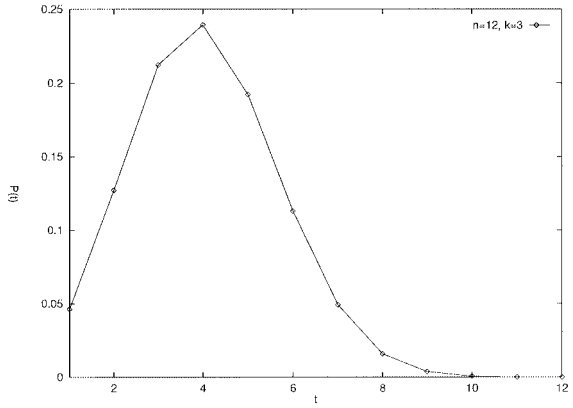

Fig. 3 Probability density function of number of tied winners.

4. Estimation

As we have discussed previously, this auction protocol has a trade-off. 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 it the purpose of an auction to discover the probability distribution on 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 use a series of ranges $[0, \omega_1), [\omega_1, \omega_2), \ldots, [\omega_{k_2}, \omega_{k_1}), [\omega_{k-1}, \infty)$. 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, under this assumption, it suffices to solve the problem for the case of uniform distribution.

4.1 Number of Tied Winners

Let n be the number of bidders and k be the dividing factor, this gives k slots of $[0, V/k), [V/k, 2V/k), \ldots, [(k-1)V/k, V)$ where V is the number of bidding prices.

For example, with letting $k = 5$, a set of bidding prices from \$1 to \$100 is divided into five slots as follows: $[1, 20), [20, 40), [40, 60), [60, 80), [80, 100)$. Let the highest bidder be willing to bid \$70. The first four successive slots have non-zero values, called active bids. We said the fourth slot, $[60, 80)$, is the highest. The highest slot always exists and has at least a bid up to n. The number of bids falling into the highest slot is identical to the number of winners in a tie. We denote the number by t.

Under an assumption that bidders choose their bids independently, we have the probability that a particular slot has a bid as $p = 1/k$. Given n and k, the probability that the highest bid is shared by t bidders is given by,

$$P_{n,k}(t) = \sum_{i=0}^{k-1} \left(\frac{k-i}{k}\right)^n \binom{n}{t} \left(\frac{1}{k-i}\right)^t \left(1 - \frac{1}{k-i}\right)^{n-t} \tag{2}$$

Figure 3 shows the probability density function of $P_{n,k}(t)$. The most likely number of tied winners is 4, which is approximated by $L[T] = np = n/k = 12/3 = 4$.

The condition of auction to be completed is that the highest slot has just one bid, that is, $t = 1$. We call it the auciton succeeds. By letting $t = 1$, we have the probability of success from Eq. (2) by

$$P_{success}(n,k) = \sum_{i=0}^{k-1} \left(\frac{k-i}{k}\right)^n n\frac{1}{k-i} \left(1 - \frac{1}{k-i}\right)^{n-1} \tag{3}$$

and illustrates the probability distribution given k with regards to n in Fig. 4.

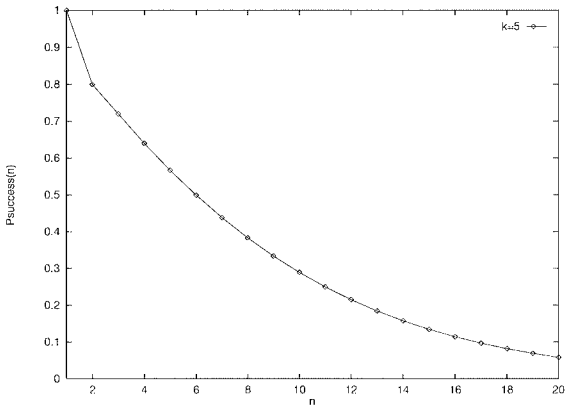

Fig. 4 Probability distribution of success.

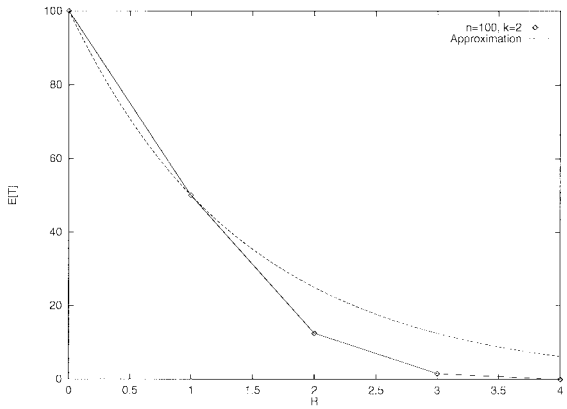

Fig. 5 Expected numbers of winners tied with.

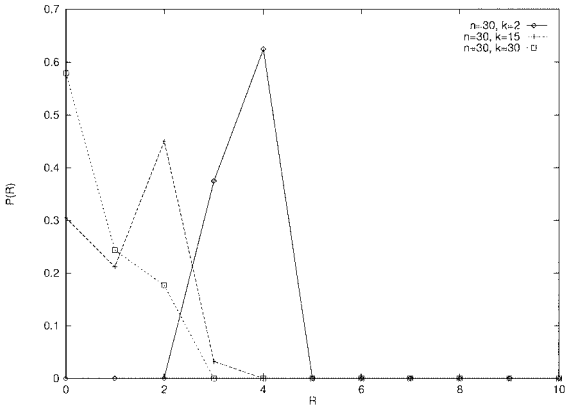

Fig. 6 Probability density function of number of rounds.

4.2 Approximation of Number of Rounds Involved

How many rounds would be involved in given n and k?

First, recall Eq. (2). It is a sum of sequential terms, which can be approximated by the first term as follows:

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

This is the Binomial distribution with $p = 1/k$, whose mean and variance are $E[T] = np$, $Var[T] = np(1 - p)$. The expected number of tied winners is np, which becomes the number of active bidders in the next round. Thus, we have the expected number of winners of i-th rounds by

$$n_i = n_{i-1}p = n_0 p^i \tag{4}$$

where $n_0 = n$, the original number of bidders at the first round. The auction ends when just one winner exists, hence, by solving $n_i = 1$, we have the expected number of rounds given n and k as follows:

$$E[R'] = \log_k(n), \tag{5}$$

where R' is a random variable representing number of rounds, which approximates the true random variable of number of rounds, R, that is, $R \simeq R'$.

4.3 Expected Number of Rounds Involved

If we directly obtain this from Eq. (2), a resolution of the expected number of rounds would be more complicated. First, We estimate the number of tied winners by $E[T] = \sum_{t=1}^{n_i} P_{n_i,k}(t)t$, which then would be the next round population, n_{i+1}. It decreases exponentially with the number of rounds and is close to the behavior of Eq. (4). We show the expected number of winners tied at the same value in Fig. 5 when $n = 100, k = 2$. In the figure, the approximation of tied winner, i.e., Eq. (4) is also indicated.

Next, recalling Eq. (3), the probability of success, we have the probability how many rounds are likely to complete the auction for given n and k by

$$P_{n,k}(r) = P_{success}(n_{r-1}, k) = P_{success}(E[N_{r-1}], k),$$

where N_r is a random variable taking the population of r-th round. Figure 6 illustrates the probabilities with regards to the number of rounds, r. Note that the most likely number of rounds for $k = 2, 15, 30$ is equal to

$$L[R] = \lceil \log_k(n) \rceil.$$

Finaly, we have the expected number of rounds given n and k by

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

Figure 7 shows an expected value of R when $n = 10$. We also indicate the previous result of the approximation, $E[R'] = \log_k(n)$, which is close to the exact result. When $n = k$, the approximation gives just one round, while the exact expected number of rounds is 0.607, which is lower than 1 in the approximation.

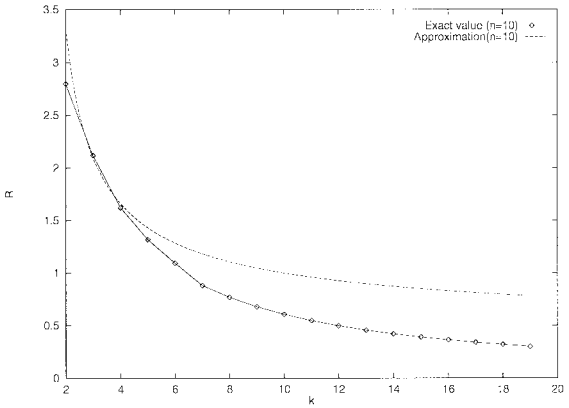

Fig. 7 The expected numbers of rounds with regards to k.

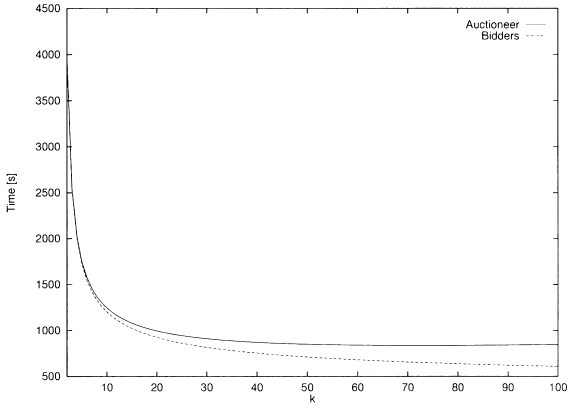

Fig. 8 Time to complete auction with regards to k.

4.4 Communication Cost

We consider the total bandwidth spent in the hierarchical auction and try to figure out the optimum dividing factor, k, given n, m, and bandwidth among them. Let C_A and C_B be bandwidth provided at auctioneers and at bidders. We assume auctioneers (server) has a higher bandwidth than bidders side. A bidder sends $Pm(k-1)$ bits per round in a channel of C_B bps, where P is a size of prime. Note that to reduce the communication, we use $(k-1)$ instead of k, because submitting no bid can be implicitly treated as bidding the lowest slot. On the other hand, an auctioneer receives Pn/C_A bits per round in a channel of C_A bps. Other than these overheads, we take into account of a period of time to synchronize all communications for each round. That takes enormous amount of time, say L sec. The expected number of rounds decreases as the dividing factor, k, increases, which makes the cost per round greater. We use the result of the approximation, Eq. (5), to estimate an expected value of rounds. The total time to complete the auction taken by an auctioneer, $T_A(k)$, is given by,

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

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

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

By differentiating both sides with k, we have

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

which can be simplified as

$$\frac{LC_A}{Pn} = k\log k\log n - k + 1. \tag{6}$$

The dividing factor k which satisfies this equation minimizes the total time to complete the auction. In the same way, the optimum k in terms of time delay at bidders can be also derived.

In Fig. 8, we show a particular behavior of $T_A(k)$ and $T_B(k)$ for the following parameters;

$$\begin{array}{ll} L = 240s & P = 100\,\text{bits} \\ C_A = 10\,\text{Mbps} & n = 10^5 \\ C_B = 28.8\,\text{kbps} & m = 10 \end{array}$$

With these constants, the time delay of auctioneer is greater than that of bidders, so the bottleneck of time delay is at the auctioneers. By solving Eq. (6), we figure out the following k^* at which the time delay of auctioneer, $T_A(k^*)$, minimizes;

$$k^* = 72.72.$$

which results in about 14 minutes to complete auction.

4.5 Normally Distributed Bids

In the above estimate, we assumed the bidders behaviors are independent, and the bids are uniformly distributed. Now, consider a normal distribution, $N(\mu, \sigma)$, with a mean μ and a standard deviation σ such that $\mu = V_r/2$, and $\sigma = \varepsilon V_r$, where ε is a constant. We denote by V_r the size of the interval of bidding values at r-th round, that is, $V_{r+1} = V_r/k = 1/k^r V_0$. While the variance and mean are setting up for each round, the fraction of the bids in the highest slot is constant. Hence, a number of tied winners at r-th round, n_r can be represented recursively by

$$n_{r+1} = \alpha_k n_r$$

where α_k is a constant assigned with given k.

For example, letting $n = 1000$, $V = 100$, and $\varepsilon = 1/6$, suppose that 1000 bids are normally distributed with the mean of $\mu = 100/2 = 50$ and the standard deviation of $\sigma = 100\varepsilon$. As commonly known property of normal distribution, a probability that a bid will exceed a particular value is characterized by the standard deviation and the mean value. When

Table 1 Probability that a bid falls in the highest slot.

k	size per slot	$\mu\pm$	α_k
2	3	0	$5 \cdot 10^{-1}$
3	2	1.0σ	$1.59 \cdot 10^{-1}$
4	1.5	1.5σ	$6.68 \cdot 10^{-2}$
5	1.2	1.8σ	$3.59 \cdot 10^{-2}$
6	1	2σ	$2.28 \cdot 10^{-2}$

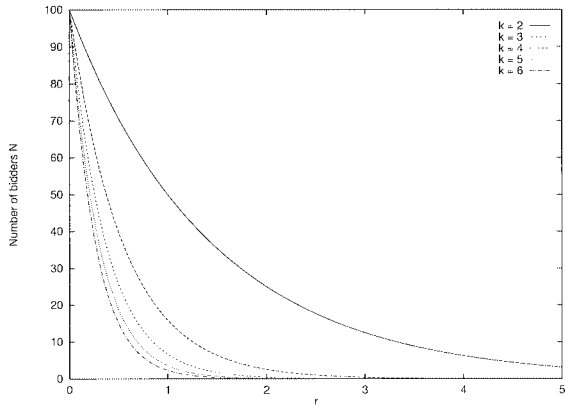

Fig. 9 Normally distributed bidders with respect to rounds.

$k = 2$, the half of the bids fall in the highest slot and the probability that a random variable of bid, B, becomes greater than $1/2$ is $P(1/2 < B) = 0.5$ If $k = 3$, the highest slot begins from $l - l/k$, which has a distance from the mean by $V - V/k - \mu = V/2 - V/3 = V/6$. We have $P(l/6/\sigma < B) = P(1.0 < B) = 0.157$, which implies that 157 bidders out of 1000 are tied with the same highest bid in average.

In generally, the constant, α_k, are given by Table 1. Figure 9 illustrates how the number of tied bidders decreases as more rounds are used. Given k and n, we estimate the number of rounds by

$$r = \log(n^{-1})/\log(\alpha_k).$$

Obviously, the smaller k than for the uniform distribution would optimize the communication cost, that is, $k = 2$.

5. Conclusion

We studied the secret bid multiple-rounds auction styles. Our proposed protocol achieves anonymity of bidders with low communication cost. From the view point of communication efficiency, we studied the probabilistic properties of the divining factor, k, and the number of rounds, r and clarified several useful propeties about bids. Our model of bids are based on independent bidders, furthermore, we examined a normally distributed bids model. The main result is that the optimum dividing factor is $k = 2$ when we assume independence of bids.

References

[1] D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," Advances in Cryptology -CRYPTO'97, Springer-Verlag, pp.425–439, 1997.
[2] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway, "Security with Low communication overhead," Crytpo '90, pp.62–76, 1990.
[3] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," STOC88, pp.1–10, 1998.
[4] D. Beaver, S. Michali, and P. Rogaway, "The round complexity of secure protocols," STOC, pp.503–513, 1990.
[5] D. Beaver and S. Goldwasser, "Multiparty computation with faulty majority," Proc. of FOCS, pp.468–473, 1989.
[6] D. Chaum, C. Crepeau, and I. Damgard, "Multiparty unconditionally secure protocols," STOC88, pp.11–19, 1988.
[7] D. Chaum, I. Damgard, and J. van de Graaf, "Multiparty computations ensuring privacy of each Party's input and correctness of the result," CRYPTO '87, LNCS 298, pp.87–119, 1987.
[8] D. Clark, "Internet cost allocation and pricing," Internet Economics, MIT Press, 1997.
[9] D. Chaum, "The dining cryptographer problem: Unconditional sender and receiver untraceability," Journal of Cryptology, vol.1, no.1, pp.65–75, 1988.
[10] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," CACM, vol.28, no.6, pp.637–647, 1985.
[11] M.K. Franklin and M.K. Reiter, "The design and implementation of a secure auction service," IEEE Trans. Software Engineering, vol.22, no.5, pp.302–312, 1996.
[12] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," Proc. Crypto '86, LNCS 263, pp.186–194, 1986.
[13] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority," ACM STOC, pp.218–229, 1987.
[14] L. McKnight and J. Bailey, "Internet Economics," MIT Press, 1997.
[15] P. Milgrom, "Auctions and bidding: A primer," Journal of Economic Perspectives, vol.3, no.3, pp.3–22, 1989.
[16] R.P. McAfee and J. McMillan, "Auctions and bidding," Journal of Economic Literature, vol.25, pp.699–738, 1987.
[17] A. Mas-Colell, M.D. Whinston, and J.R. Green, "Microeconomic Theory," Oxford university press, pp.857–925, 1995.
[18] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," STOC '89, pp.73–85, 1989.
[19] A. Shamir, "How to share a secret," CACM, vol.22, pp.612–613, 1979.
[20] A.C. Yao, "Protocols for secure computations," Proc. 27th IEEE Symposium on Foundations of Computer Science, pp.162–167, 1986.

Hiroaki Kikuchi was born in Japan. He received B.E. , M.E. and Dr.E. degrees from Meiji University in 1988, 1990 and 1994. He joined Fujitsu Laboratories Ltd. in 1990. He was a visiting researcher of school of computer science, Carnegie Mellon university in 1997. He is currently an Assistant Professor in Department of Electrical Engineering, Faculty of Engineering, Tokai University. His main research interests are fuzzy logic and network security. Dr. Kikuchi is a member of the Information Processing Society of Japan, and the Japan Society for Fuzzy Theory and Systems.

Michael Hakavy is a Ph.D. student in the School of Computer Science at Carnegie Mellon Univeristy. He is currently on leave of absence and working at University of California, Berkeley.

Doug Tygar was born and raised in the San Francisco Bay Area. He received his bachelor's degree from the University of California, Berkeley in 1982, and a Ph.D. in computer science in 1986 from Harvard University. After receiving his Ph.D., Doug Tygar has been at Carnegie Mellon University, where he is a tenured faculty member in computer science. Doug Tygar is a full professor at UC Berkeley. He holds a joint appointment 50 Engineering and Computer Science and 50. He is extremely active in the electronic commerce and computer security communities.