A New Efficient Polynomial Degree Resolution Protocol and Its Applications to the (M+1)-st Price Private Auction

Anthony T. Chronopoulos[1], Daniel Grosu[2], and Hiroaki Kikuchi[3]

[1] Dept. of Computer Science, Univ. of Texas at San Antonio, 6900 N. Loop 1604 West, San Antonio, TX 78249, U.S.A. atc@cs.utsa.edu,

[2] Dept. of Computer Science, Wayne State University, 5143 Cass Avenue, Detroit, MI 48202,U.S.A. dgrosu@cs.wayne.edu,

[3] Dept. of Electrical Engineering, Tokai University, 1117 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan. kikn@ep.u-tokai.ac.jp

Abstract. Cryptographic protocols have been proposed and studied for application to electronic auctions in the past. One such protocol is the (M+1)-st price private auction. In this protocol the highest M bidders win and pay a uniform price determined by the (M+1)-st highest bid. The (M+1)-st price is determined by a set of distributed servers who collaborate while keeping the bids secret. The highest bid is represented by the degree of an interpolation polynomial in Lagrange form. In this article we study the computational efficiency of this approach and we propose the use of Newton form of interpolation. This approach is computationally efficient providing a significant reduction in the number of operations required for solving the auction.

1 Introduction

The study in this paper is motivated by the recent increased interest in designing auction protocols providing privacy of bids. In this paper, we consider a secure protocol for sealed-bid $(M+1)$-st price auction. In this type of auction multiple units of a single item are auctioned. The M highest bidders win and pay a uniform price, the price of $(M+1)$-st highest bidder.

By letting M be 1, the definition includes as a special case the second-price auction, or the so called Vickrey auction [12]. Wurman *et al.* [23] proved that the $(M+1)$-st price auction satisfies a useful property called *incentive compatibility*, i.e., the dominant strategy for a bidder is to bid his/her true valuation, as it is well known for the widely advocated Vickrey auction. Since the winners' payments will be determined by the $(M+1)$-st highest bid, every bidder who agrees to bid the maximum price he/she is willing to pay for a given item maximizes his/her chance to win without being worried that he/she might bid too much. Furthermore, the sealed-bid auction is fast because all that bidders have to do is to cast their sealed bids just once. The large amount of interaction between auctioneers and bidders in an on-line auction, is not necessary in a sealed-bid auction.

Related work

Franklin and Reiter presented a sealed-bid auction protocol in [4]. The protocol uses a verifiable signature sharing in order to prevent malicious bidders from canceling their bids. Bids are kept secret until the opening phase, and then all bids are opened and compared to determine the highest one. Kikuchi, Harkavy and Tygar [6] improved the privacy of bids among distributed auctioneers even after the opening phase using a secure function computation of summation. The protocols run in linear time to the number of possible bidding prices and cannot deal with tie breaking.

Any Dutch-style auction naturally satisfies the property that privacy of losing bids is preserved after auction closes. In [18, 19], Sako implemented a Dutch-style auction using distributed decryption. In the protocol, a bidder casts his bid encrypted by the public key corresponding to his bidding price. The privacy of losers' bids is kept under the assumption that not all the auctioneers are faulty. Similarly, Miyazaki and Sakurai used undeniable signatures [13], and Kobayashi and Morita use an one-way hash chain [9]. A cryptographic Vikrey auction scheme that involves an auction authority is proposed in [11]. Recently, several works on private auctions have been published [3, 10, 16, 20–22].

Auctions in electronic commerce are more complicated. Multiple buyers and sellers are involved and multiple unit of goods are auctioned in several environments. Wurman, Walsh and Wellman [23] studied several auction designs and analyzed them in terms of incentive compatibility. They showed that the $(M+1)$-st price sealed-bid auction is incentive compatible for single-unit buyers. A secure second-price ($M = 1$) auction protocol is presented by Harkavy *et al.* in [5]. They used the secure multiparty protocol for multiplication, presented in [2] in order to resolve the second highest bid in $O(\log(k))$ rounds, where k is the number of possible bid values. Recently, Miyamoto *et al.* [14] implemented this protocol but due to the communication cost among auction servers, $O(n)$, an enormous amount of time is required to decide the winning price. Kikuchi presented a more general protocol for $(M + 1)$-st price auction in [7]. The protocol, however, is definitely inefficient because it takes a cost of n-choose-k and has a serious security flaw.

Our contributions

We propose an efficient degree resolution protocol used for resolving the (M+1)-st price in the (M+1)-st price private auction. The protocol is based on the Newton interpolation which is more efficient than the previously used Lagrange form [8] because it is computed iteratively. Thus, it reuses the preceding degree computations when the polynomial degree increases or decreases. This implies that the new protocol is more efficient, significantly reducing the number of operations required for degree resolution.

Organization

The paper is structured as follows. In Section 2 we describe the model and formulate the problem. In Section 3 we give a brief review of interpolation methods. In Section 4 we describe the interpolation based degree resolution methods. In Section 5 we describe our improved auction protocols. In Section 6 we draw conclusions and present future research directions.

2 Model and problem formulation

Given M units of a single good, n bidders are going to buy goods at a uniform price, which is determined in a meaningful procedure. Let $W = \{w_1, \ldots, w_k\}$ be a set of k possible discrete bidding prices. The i-th bidder has his/her true evaluation $e_i \in W$. The objective of the auction game is to find the $(M + 1)$-st highest price w^* of all bids without revealing any bids, even those higher than w^*, and to find M winners who have bids higher than w^*.

We assume that each bidder has independent private evaluation for goods. The evaluation, e_i, is not affected by the evaluations other bidders place on the good. This is called the private values model and it is widely used when modeling auctions. In the theory of economics, it is known that a social surplus is maximized when bidders whose bid is higher than w^* win the

auction game and pay the uniform winning price which is independent of their evaluation. The Vickery auction, in which the winner who has the highest bid pays the second highest bid, is a special case with $M = 1$.

In our model we consider m auctioneers that collaborate to resolve the winning price in such a way that no c auctioneers can be faulty. Auctioneers are m independent servers. Bidders do not trust each of the auctioneers, but trust an agreement of more than c auctioneers. Auctioneers do not trust bidders, who might violate the specified protocol in order to disrupt the auction. The protocol is based on the information-theoretic secure verifiable sharing protocol proposed in [17]. We assume confidentiality of every session, entity authentication, and integrity of messages based on appropriate cryptographical tools including PKI. Hence, eavesdropping links give no information about bids or bidders.

Requirements

Privacy of bid: No bid is revealed to anyone except the $(M + 1)$-st highest bid. For the sake of the incentive compatibility, we want to make leakage of information as small as possible. Thus, even the bids higher than the winning bid must be secret even after the auction closes. No statistics can be used to identify the distribution of bids even after the auction closes.

Proof of winner: The winner must publicly prove that his/her bid is higher than the winning bid without revealing how high the bid is.

Non-repudiation: No bidder can repudiate his bid. If bidders are allowed to cancel their bids, a collusion of malicious bidders can control the winning price as they like (this attack was mentioned first in [15]).

Accountability of bidder: Any auctioneer can verify that bidders follow a protocol to cast their bids. No malicious bidder can disrupt the auction with an unmannered bid without being detected.

Accountability of auctioneer: Any bidder can verify that auctioneers correctly follow a protocol to resolve the winning bid. No malicious auctioneer can alter the result of auction without being detected.

Round efficiency: The protocol is efficient in terms of rounds involved in resolving the winner.

Communication efficiency: The protocol is efficient in terms of bandwidth consumption between bidders and auctioneers. The communication among servers must be minimized.

3 Review of Polynomial Interpolation

We consider a function $f(x)$ with (input/output) values in a finite field. We assume given interpolation nodes, $x_1 < \ldots < x_n$ and the values of the function at the nodes, $f_i = f(x_i)$, for $i = 1, \ldots, n$.

3.1 The Lagrange Interpolation Polynomial

The Lagrange (interpolation) polynomial (of degree at most n) is defined as [1]:

$$Pf(x) = \sum_{j=1}^{n} f(x_j) L_j(x) \tag{1}$$

where the polynomial matches the function values at the nodes: $Pf(x_i) = f(x_i)$ for $i = 1, \ldots, n$ and $L_j(x)$ are the Lagrange interpolation coefficients defined to give: $L_j(x_i) = 0$, when $i \neq j$ and $L_j(x_i) = 1$, when $i = j$. This implies that

$$L_j(x) = \frac{(x - x_1) \ldots (x - x_{j-1})(x - x_{j+1}) \ldots (x - x_n)}{(x_j - x_1) \ldots (x_j - x_{j-1})(x_j - x_{j+1}) \ldots (x_j - x_n)} = \frac{\prod_{i \neq j}(x - x_i)}{\prod_{i \neq j}(x_j - x_i)} \tag{2}$$

We can now rewrite $Pf(x)$ as

$$Pf(x) = \prod_{i=1}^{n}(x - x_i) \sum_{j=1}^{n} \frac{f(x_j)}{(x - x_j) \prod_{i \neq j}(x_j - x_i)} \tag{3}$$

Now we can write an efficient algorithm to compute $Pf(x)$.

Algorithm for computing $Pf(x)$ using Lagrange Interpolation:

Step 1: $\psi_j = \frac{f(x_j)}{\prod_{i \neq j}(x_j - x_i)}, \qquad j = 1, \ldots, n$
Step 2: $\phi(x) = \prod_{i=1}^{n}(x - x_i)$
Step 3: $Pf(x) = \phi(x) \sum_{j=1}^{n} \frac{\psi_j}{(x - x_j)}$

We note that Step 1 is a preprocessing step and it is executed only once for the given nodes. Steps 2 and 3 are executed once per computation of $Pf(x)$.

The complexity of Lagrange interpolation algorithm:

In order to analyze the complexity of the interpolation algorithm we consider that the cost of multiplication operations is higher than the cost of subtraction and addition operations. We also consider the cost of multiplications equal to the cost of obtaining the multiplicative inverses. In the following analysis an operation is considered to be the multiplication operation. The total number of operations in Step 1 is $n(n-1)$, in Step 2 is $n-1$, and in Step 3 is $n+1$. The total number of operations involved in the computation of the Lagrange interpolation polynomial $Pf(x)$ with n nodes is $n^2 + n$. The cost to evaluate the Lagrange polynomial Pf at k different values x is $(n^2 - n) + k(2n)$.

3.2 The Newton Interpolation Polynomial

Definition 1. (Divided Differences) Let $x_1, x_2, \ldots, x_n$ be distinct elements in a finite field. The zero order divided difference is defined as: $DF(0) = f(x_1)$. The first order divided difference is defined as: $DF(1) = f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$. The $k-1$ order divided difference is defined as:

$$DF(k-1) = f[x_1, \ldots, x_k] = \frac{f[x_2, \ldots, x_k] - f[x_1, \ldots, x_{k-1}]}{x_k - x_1} \tag{4}$$

Definition 2. (Newton Divided Difference Interpolation) [1] Let $Qf(x)$ be a polynomial of degree (at most) n,

$$Qf(x) = f[x_1] + (x - x_0)f[x_1, x_2] + \ldots + (x - x_1)(x - x_2) \ldots (x - x_{n-1})f[x_1, \ldots, x_n] \tag{5}$$

This implies that we must compute once the (triangular) array of divided differences and then we can evaluate the Newton polynomial $Qf(x)$ at different input values x.

Remark: The Newton polynomial has a recursive property. Let $Qf_n(x)$ be the Newton polynomial based on nodes $x_1, \ldots, x_n$ and $Qf_{n-1}(x)$ be the Newton polynomial based on nodes $x_1, \ldots, x_{n-1}$ then it easy to check that

$$Qf_n(x) = Qf_{n-1}(x) + (x - x_1)(x - x_2)\ldots(x - x_{n-1})f[x_1, \ldots, x_n]. \quad (6)$$

Horner's rule is often used to evaluate the Newton polynomial. For example for the two degree case: $Qf(x) = DF(0) + (x - x_1)(DF(1) + (x - x_2)DF(2))$ We note that the main steps in computing the Newton polynomial are: 1. Compute (once) the array of the divided differences. 2. For each x compute $Qf(x)$ using Horner's rule.

Algorithm for computing $Qf(x)$ using Newton Interpolation:

A1. Compute the finite differences

for $j = 1, \ldots, n-1$ **do**
  $DF(j) = f(x_{j+1})$
for $i = 1, \ldots, n-1$ **do**
  for $j = n-1, \ldots, i$ **do**
    $DF(j) = \frac{DF(j)-DF(j-1)}{x_j - x_{j-i}}$

A2. Evaluate the polynomial at input x, using Horner's rule

Initially: $Qf(x) = DF(n-1)$
for $i = n-2, \ldots, 1$ **do**
    $Qf(x) = DF(i) + (x - x_i)Qf(x)$

The complexity of Newton interpolation algorithm:

Using the same assumptions from the complexity analysis of Lagrange interpolation we get the following. The total number of operations involved in the computation of the n-th Newton interpolation is $\frac{n(n-1)}{2} + n - 2 = (1/2)n^2 + (1/2)n - 2$. The cost to evaluate the Newton polynomial Qf at k different values x is $(1/2)(n^2 - n) + k(n-2)$.

Remarks: (i) The Newton and Lagrange methods compute the same interpolation polynomial over finite fields. The polynomial $Rf(x) = Pf(x) - Qf(x)$ has roots at the nodes $x_1, \ldots, x_n$. This implies that on any field $Rf(x) = 0$. (ii) Compared to Lagrange interpolation there are about 50% fewer operations. (iii) The recursive property of the Newton polynomial allows the computation of the one degree higher polynomial from the current degree polynomial with 2 subtractions, 1 division and 1 multiplication operation. We must store the main diagonal of the triangular array of divided differences.

4 Polynomial Degree Resolution using Interpolation

We denote by $f^{(s)}(0)$ the s-th interpolation (Lagrange or Newton) of f. We can learn the degree of f as the least s that satisfies $f^{(s)}(0) = f(0)$. The degree resolution succeeds with probability $1/p$, assuming random picking for α_i, where $\alpha_1, \alpha_2, \ldots, \alpha_s \in Z_p^*$ are the interpolation nodes.

Also, we can learn the degree of f given the values $g^{f(\alpha_1)}, g^{f(\alpha_2)}, \ldots, g^{f(\alpha_t)}$ as the least s that satisfies:

$$g^{f(0)} = g^{f^{(s)}(0)} = g^{f^{(s-1)}(0)} g^{(-1)^{s-1} DF(s-1)} \quad (7)$$

Assuming random picking from Z_p^* gives $f(0)$ with probability of $1/p$, we have the probability that the degree resolution mistakenly succeeds as follow. Probability of $t < s$ given $f^{(s)}(0) = f(0)$ is $1 - 1/p$. Note that the probability can be negligible with p increasing.

4.1 Degree Resolution using the Lagrange Interpolation Polynomial

Definition 3. (Lagrange Interpolation) Let f be a polynomial of degree t, $f(x) = a_0 + a_1 x + \ldots + a_t x^t$. We denote by $f^{(s)}(0)$ the s-th Lagrange interpolation of f, defined as follows: $f^{(s)}(0) = \sum_{j=1}^{s} f(\alpha_j) \prod_{i \neq j} \frac{\alpha_i}{\alpha_i - \alpha_j}$, where $\alpha_1, \alpha_2, \ldots, \alpha_s \in Z_p^*$ are the interpolation nodes.

Algorithm for computing $f^{(s)}(0)$ using Lagrange Interpolation:

Step 1: $\psi_j = \frac{f(\alpha_j)}{\prod_{i \neq j} (\alpha_i - \alpha_j)}, \qquad j = 1, \ldots, s$
Step 2: $\phi(0) = \prod_{i=1}^{s} \alpha_i$
Step 3: $f^{(s)}(0) = \phi(0) \sum_{j=1}^{s} \frac{\psi_j}{\alpha_j}$

The complexity of Lagrange interpolation algorithm:
Using the results of the analysis in Section 3, the total number of operations involved in the computation of the s-th Lagrange interpolation is $s^2 + s$.

4.2 Degree Resolution using Newton Interpolation

Definition 4. (Newton Divided Difference Interpolation) Let f be a polynomial of degree t, $f(x) = a_0 + a_1 x + \ldots + a_t x^t$. We denote by $f^{(s)}(0)$ the Newton interpolation of f, defined as follows: $f^{(s)}(0) = f^{(s-1)}(0) + (-1)^{s-1} DF(s-1)$, where $\alpha_1, \alpha_2, \ldots, \alpha_s \in Z_p^*$.

Algorithm for computing $f^{(s)}(0)$ using Newton Interpolation:

A1. Compute the finite differences

for $j = 1, \ldots, s-1$ **do**
    $DF(j) = f(\alpha_j)$
for $i = 1, \ldots, s-1$ **do**
    for $j = s-1, \ldots, i$ **do**
        $DF(j) = \frac{DF(j) - DF(j-1)}{\alpha_j - \alpha_{j-i}}$

A2. Compute the polynomial

for $i = 1, \ldots, s$ **do**
        $f^{(i)}(0) = f^{(i-1)}(0) + (-1)^{i-1} DF(s-1)$

The complexity of Newton interpolation algorithm:
The total number of operations involved in the computation of the s-th Newton interpolation is $(1/2)(s^2 - s) + 2s = (1/2)s^2 + (3/2)s$. Compared to Lagrange interpolation there are approximately 50% fewer operations. Another advantage of Newton algorithm is that we can compute the s-th interpolation from the $(s-1)$-th interpolation using only three operations

and the $(s-1)$ divided difference. This is a very useful property that will be used by the degree resolution algorithm in the auction protocols that we propose.
Remark: The Newton and Lagrange methods compute the same interpolation polynomial in finite fields.

The Lagrange interpolation is very inconvenient for actual calculations, especially when we interpolate with polynomials of various degree. The advantage of using Newton interpolation is that we can recursively compute $f^{(s)}(0)$ by using $f^{(s-1)}(0)$. For computing $f^{(s)}(0)$ we also need to compute the $s-1$ order divided difference.

5 Auction Protocols

We present an efficient first price auction protocol which is based on the protocol presented by Kikuchi in [8]. The protocol finds the highest price from n bids without revealing any of the bids. In the following we use the notations in [8] to describe the protocols. Here, A_j denotes the auctioneer j, $j = 1, \ldots, m$ and B_i denotes the bidder i, $i = 1, \ldots, n$.
FIRST-PRICE Protocol

Step 1 Let $b_i \in \{1, 2, \ldots, k\}$ be the bid of bidder i such that $w_{b_i} = e_i$. Bidder i randomly picks a polynomial $f_i(x) = \sum_{j=1}^{t_i} a_j x^j$. This polynomial will have degree $t_i = b_i + c$, where constant c is the number of faulty auctioneers. Bidder i sends the share $f_i(\alpha_j)$ to auctioneer j, $j = 1, 2, \ldots, m$. Note that $a_0 = 0$.

Step 2 A_j receives the n shares, $f_i(\alpha_j)$, sent by bidders and uses them to compute: $F(\alpha_j) = \sum_{i=1}^{n} f_i(\alpha_j)$. Then A_j publishes $F(\alpha_j)$ using a suitable commitment protocol.

Step 3 Any bidder or auctioneer can find the maximum bid, b^*, by using the following procedure (degree resolution using Newton interpolation):

for $i = 1, \ldots, m$ **do**
$F^{(i)}(0) = F^{(i-1)}(0) + (-1)^{i-1} F[\alpha_1, \alpha_2, \ldots, \alpha_i]$
if $(F^{(i)}(0) = 0)$ **break**;
$b^* = i - c$;

Complexity: Using Newton interpolation we are able to compute the s-th interpolation given the $(s-1)$-th interpolation. There are only a few number of operations that are involved. These are: one multiplication, one addition and the operations involved in computing the divided difference $F[\alpha_1, \alpha_2, \ldots, \alpha_i]$. To compute the divided difference we need to compute the values of all the elements of the s-th row of the divided difference matrix. For each element of this row we need two subtractions and one division. For all $s-1$ elements of the last row we need to perform $s-1$ operations (considering the assumptions from Section 3). So the total number of operations needed to obtain the s-th interpolation from the $(s-1)$-th interpolation is $s-1$.

For the Lagrange algorithm the total number of operations to compute the s-th interpolation is $3s+1$. Considering the Lagrange algorithm presented in the previous section we can determine the number of operations. We assume that the results form $(s-1)$-th interpolation are stored and used to compute the s-th interpolation. The total number of operations in Step 1 is $3s-2$. Step 2 requires only one operation and Step 3 requires two operations. Thus the

total number of operations is: $3s-2+1+2 = 3s+1$. Using the Newton algorithm the number of operations is reduced from $3s+1$ to $s-1$ which is about 66% reduction.

The above protocol assumes that the participants are honest. To make this protocol more realistic we need to handle malicious bidders and auctioneers.

5.1 New Verification Protocol

The following protocol is an improved protocol for the first price auction which prevents both malicious bidders and auctioneers from misbehaving.

VFIRST-PRICE Protocol

Step 1: B_i chooses two random polynomials: $f_i(x) = \sum_{j=1}^{t_i} a_j x^j$ of degree $t_i = b_i + c$ and $h_i(x) = \sum_{j=1}^{s} b_j x^j$ of degree $s = k+c$ with $s > t_i$. B_i secretly sends $f_i(\alpha_j)$ and $h_i(\alpha_j)$ to A_j, for $j = 1, \ldots, m$. Then, B_i publishes the following values as commitments of polynomials: $E_{i,1} = g_1^{a_1} g_2^{b_1}, \ldots, E_{i,t_i} = g_1^{a_{t_i}} g_2^{b_{t_i}}, E_{i,t_i+1} = g_2^{b_{t_i+1}}, \ldots, E_{i,s} = g_2^{b_s}$.

Step 2: A_j verifies that the share sent from bidder i is consistent with the commitments as, $g_1^{f_i(\alpha_j)} g_2^{h_i(\alpha_j)} = X_{i,j} = \prod_{l=1}^{s} (E_{i,l})^{\alpha_j^l}$. If the identity holds, she is convinced that $f_i(x)$ has no constant ($a_0 = 0$) and it is of degree of at most s, and then publishes: $Y_j = g_1^{F(\alpha_j)}$ and $Z_j = g_2^{H(\alpha_j)}$, where $F(\alpha_j) = f_1(\alpha_j) + , \ldots, + f_n(\alpha_j)$ and $H(\alpha_j) = h_1(\alpha_j) + , \ldots, + h_n(\alpha_j)$.

Step 3: Any entity can verify that Y_j and Z_j are computed correctly by testing $Y_j Z_j = \prod_{i=1}^{n} X_{i,j} = g_1^{F(\alpha_j)} g_2^{H(\alpha_j)}$. If this holds then the highest price is given by $b^* = t^* - c$ where $t^* \in \{1, \ldots, k\}$ is obtained by the following procedure (degree resolution based on Newton interpolation):

for $i = 1, \ldots, m$ **do**
$(g_1^F)^{(i)}(0) = (g_1^F)^{(i-1)}(0) + (-1)^{i-1}(g_1^F)[\alpha_1, \alpha_2, \ldots, \alpha_i]$
if $((g_1^F)^{(i)}(0) = 0)$ **break**;
$b^* = i - c$;

Complexity: The same comments from the previous algorithm apply here. The difference in this algorithm is that we use exponentiation which is more expensive than the basic arithmetic operations. So the reduction in the execution time will be more significant.

In this protocol, neither the bidders nor the auctioneers can cast bogus values without being detected. The protocol is still vulnerable to the winner attack, which will be solved in the next section.

5.2 Finding the Winners

The following protocol is used to determine the winning price and the winner's identities.

WINNER Protocol

Step 1: B_i chooses three random polynomials: $f_i(x) = \sum_{j=1}^{t_i} a_j x^j$, $h_i(x) = \sum_{j=1}^{s} c_j x^j$ and $G_i(x) = \sum_{j=1}^{s-t_i} b_j x^j$. B_i secretly sends $f_i(\alpha_j)$, $G_i(\alpha_j)$ and $h_i(\alpha_j)$ to A_j, for $j = 1, \ldots, m$. Then, B_i publishes the following values: $E_{i,l} = g_1^{a_l b_l} g_2^{c_l}$ for $l = 1, \ldots, s$.

Step 2: A_j verifies that the share sent from bidder i is consistent with the commitments as, $g_1^{f_i(\alpha_j)G_i(\alpha_j)} g_2^{h_i(\alpha_j)} = X_{i,j} = \prod_{l=1}^{s} (E_{i,l})^{\alpha_j^l}$. If the identity holds, she is convinced that $f_i(x)$ has no constant ($a_0 = 0$) and it is of degree of at most s, and then publishes: $Y_j = g_1^{F(\alpha_j)}$ and $Z_j = g_2^{H(\alpha_j)}$, where $F(\alpha_j) = f_1(\alpha_j) + , \ldots, + f_n(\alpha_j)$ and $H(\alpha_j) = h_1(\alpha_j) + , \ldots, + h_n(\alpha_j)$. Note that, unlike the VFIRST-PRICE, shares $G_1(\alpha_j), \ldots, G_n(\alpha_j)$ are kept secret locally at this point.

Step 3: Any entity can verify that Y_j and Z_j are computed correctly by testing $Y_j Z_j = \prod_{i=1}^{n} X_{i,j} = g_1^{F(\alpha_j)} g_2^{H(\alpha_j)}$. If this holds then the highest price is given by $b^* = t^* - c$ where $t^* \in \{1, \ldots, k\}$ is obtained by the following procedure:

for $i = 1, \ldots, m$ **do**
$(g_1^F)^{(i)}(0) = (g_1^F)^{(i-1)}(0) + (-1)^{i-1}(g_1^F)[\alpha_1, \alpha_2, \ldots, \alpha_i]$
if $((g_1^F)^{(i)}(0) = 0)$ **break**;
$b^* = i - c$;

Then a subset of auctioneers whose size is $u = s - t^*$ collaborate to resolve winners by revealing a sequence of shares $G_1(\alpha_j), \ldots, G_n(\alpha_j)$ for $j = 1, \ldots, u$. There must be (at least one) bidder i^* such that $G_{i^*}^{(u)}(0) = 0$, which proves his bid is the highest. They execute the following procedure to verify the above equation:

for $i = 1, \ldots, u$ **do**
$G_{i^*}^{(i)}(0) = G_{i^*}^{(i-1)}(0) + (-1)^{i-1} G_{i^*}[\alpha_1, \alpha_2, \ldots, \alpha_i]$
if $(G_{i^*}^{(i)}(0) = 0)$ **then** B_{i^*} is the winner.

Complexity: In this algorithm we use the Newton interpolation algorithm twice. So we will save more operations than in the previous protocol.

5.3 Simple (M+1)st-Price Auction

To extend the first-price auction to $(M+1)$st-price, the simplest way is to iterate Protocol WINNER excluding the winner i^* from the set of bidders as $Y_j^{(l)} = \frac{Y_j^{(l-1)}}{g_1^{f_{i^*}(\alpha_j)}}$, for $l = 1, \ldots, M$ and $j = 1, \ldots, m$. Let $Y_j^{(0)} = Y_j$ at Step 3 in the WINNER protocol. After M winners are determined, the set of auctioneers use Protocol WINNER to identify the $(M+1)$-st price, say t^*, while keeping the $(M+1)$-st highest bidder anonymous.

Remark: Protocol WINNER determines a set of winners without revealing losers' bids.

Unless more than t^* auctioneers collude and leak the corresponding $G_i(\alpha)$, the privacy of $(M+1)$-st highest bidder is preserved. The protocol, however, reveals all winners' private bids, which are not required because the winners pay the uniform price, $t^* - c$.

6 Conclusion

We presented a new efficient degree resolution protocol used in the (M+1)-st price private auction. This protocol is based on Newton polynomial interpolation. This approach significantly reduces the number of operations needed to determine the highest bid compared to

the existing approach. Future work will address the implementation of an auctioning system based on this new efficient protocol.

Acknowledgments: This research was supported, in part, by NSF grant CCR-0312323 and by a grant from the Center for Infrastructure Assurance and Security at The University of Texas at San Antonio.

References

1. K. Atkins, *Elementary Numerical Analysis*, John Wiley & Sons, New York, 1993.
2. M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, In *Proc. of the 20th Annual ACM Symp. on Theory of Computing (STOC'88)*, pp.1-10, May 1988.
3. C. Cachin, Efficient private bidding and auctions with an oblivious third party, In *Proc. of the 6th ACM Conference on Computer and Communications Security*, pp.120-127, November 1999.
4. M. K. Franklin and M. K. Reiter, The design and implementation of a secure auction service, *IEEE Trans. on Software Engineering*, 22(5), pp. 302-312, 1996.
5. M. Harkavy, J. D. Tygar, and H. Kikuchi, Electronic auction with private bids, In *Proc. of the 3rd USENIX Workshop on Electronic Commerce*, pp.61-74, August 1998.
6. H. Kikuchi, M. Harkavy and J. D. Tygar, Multi-round anonymous auction, *IEICE Transactions on Information and Systems*, E82-D(4), pp.769-777, 1999.
7. H. Kikuchi, Power auction protocol without revealing bidding prices, In *Proc. of SCIS'00*, 2000, pp.B10, (in Japanese).
8. H. Kikuchi, (M+1)st-Price Auction, In *Proc. of the 5th International Conference on Financial Cryptography*, pp. 291-298, February 2001.
9. K. Kobayashi and H. Morita, Efficient sealed-bid auction with quantitative competition using one-way functions, In Technical Report of IEICE, ISEC99-30, pp.31-37, 1999.
10. M. Kudo, Secure electronic sealed-bid auction protocol with public key cryptography, *IEICE Transactions on Fundamentals*, E81-A(1), pp.20-26, 1998.
11. H. Lipmaa, N. Asokan and V. Niemi, Secure Vickrey Auctions without Threshold Trust, In *Proc. of the 6th International Conference on Financial Cryptography*, March 2002.
12. P. Milgrom, Auctions and bidding: a primer, *Journal of Economic Perspectives*, 3(3), pp.3-22, 1989.
13. S. Miyazaki and K. Sakurai, A bulletin board-based auction system with protecting the bidder's strategy, *Transactions of IPSJ*, 40 (8), pp.3229-3336, 1999 (in Japanese).
14. M. Miyamoto, H. Kikuchi and K. Ogino, Dispersive auction server to calculate only the second win-price keeping the bids secret, In *Proc. of DICOMO 2000 Symposium*, IPSJ Symposium Series 2000 (7), pp.547-552, 2000 (in Japanese).
15. T. Nakanishi, H. Watanabe, T. Fujiwara and T. Kasami, An Anonymous Bidding Protocol Using Undeniable Signature, In *Proc. of the 1995 Symposium on Cryptography and Information Security*, pp.B1.4, 1995 (in Japanese).
16. M. Naor, B. Pinkas and R. Sumner, Privacy preserving auctions and mechanism design, In *Proc. of the 1st ACM Conference on Electronic Commerce*, pp. 129-139, November 1999.
17. T. P. Pedersen, Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing, In *CRYPTO'91*, pp.129-140, 1992.
18. K. Sako, An auction protocol which hides bids of losers, In *Proc. of the Intl. Workshop on Practice and Theory of Public Key Cryptography (PKC'2000)*, pp.422-432, 2000.
19. K. Sako, Universal verifiable auction protocol which hides losing bids, In *Proc. of the 1999 Symposium on Cryptography and Information Security*, pp.35-39, 1999 (in Japanese).
20. F. Stajano and R. Anderson, The cocaine auction protocol: on the power of anonymous broadcast, In *Proc. of Information Hiding Workshop 1999*, (LNCS), 1999.
21. S G. Stubblebine and P. F. Syverson, Fair On-Line Auctions without Special Trusted Parties, in *Proc. of Financial Cryptography 1999*, LNCS 1648, 1999, pp.230-240.
22. Y. Watanabe and H. Imai, Optimistic Sealed-Bid Auction Protocol, In *Proc. of the 2000 Symposium on Cryptography and Information Security*, B09, pp.1-8, 2000.
23. P. R. Wurman, W. E. Walsh and M. P. Wellman, Flexible Double Auctions for Electronic Commerce: Theory and Implementation, *Decision Support Systems*, 24, pp.17-27, 1998.