Cryptographic Protocols for Sealed-Bid Auctions Without Trusted Servers

Wen-Guey Tzeng* · Nai-Chia Yeh** · Rong-Jaye Chen**

* Department of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan

** Department of Computer Science and Information Engineering
National Chiao Tung University, Hsinchu, Taiwan

e-mail : njyeh@csie.nctu.edu.tw

Abstract

Sealed-bid auctions are a kind of auctions that bidders send their sealed bidding prices to the auctioneer and the auctioneer opens all bids to determine the winner and the winning bid. In this paper, we propose a novel construction on the secure computation of the greater-than function. This idea is applied to the two-party private bidding and sealed-bid auction protocols. Our protocols require no trusted servers. In the private bidding protocol, it needs two rounds of communication between the bidders. In the auction protocol, a public bulletin board is used. It takes two or three rounds of communication between the bidders and the board. After the execution of our protocols, no bid information is revealed to any party.

Key words: sealed-bid auction, secure multiparty computation, Diffie-Hellman key exchange, privacy

1 Introduction

As Internet prevails, electronic commerce becomes an important issue. Traditional face-to-face communication and trading activities are translated into electronic processes, such as electronic mail, digital money, electronic voting, and so forth. Internet indeed facilitates human life, but it is an important issue how we keep personal information secret in the network. Traditional laws and contracts are not sufficient to protect our privacy. That is why we study cryptography. By cryptographic techniques, trading activities are implemented in the computer network which meets various security requirements.

Generally speaking, a trading process consists of three phases : price negotiation, payment, and goods delivery. Auctions are a form of price negotiation that allow buyers to settle the price of goods. They can roughly be classified into two types, open-cry auctions and sealed-bid auctions. In the open-cry auction, each buyer bids his price publicly and at the close of the auction the bidder who bids the highest price wins and gets the goods. The disadvantages of this type of auctions are time-consuming and revelation of the bid information of buyers. It is almost impossible to have a fair open-cry auctions in the unreliable network. Sealed-bid auctions are usually used in the auction of government procurement contracts, public construction and government-owned land etc. Each bidder seals his bidding price and submits it to the auctioneer. At the close of the auction, the auctioneer opens all bids to determine the winner and the selling price. This type of auctions is more efficient in network communication. For the reasons of security and practicability, we study the topic of sealed-bid auctions. Many researchers are devoted to the study of secure and efficient sealed-bid auctions[FR96][Sak00][HTK98][Cac99][BS][Fis]. Several approaches are based on the multiparty computation and secret sharing to ensure the privacy of bid information.

In this paper, we give a new direction to compare two secrets securely. One protocol is presented for two-party private bidding and two protocols for first-price sealed-bid auctions. In the private bidding protocol, no trusted servers are required for a fair bidding between two players. It needs only two rounds of interaction. The computation complexity of one party is 4ℓ modular exponentiations. In the auction protocols, we use a public bulletin board. One of our auction protocols requires two rounds of communication, and the other requires three rounds. During each round of our protocols, each bidder posts some messages on the board. At the end of the auction, each bidder's bid information is unknown to others. Only the winning bid is revealed to all participants by the winner.

2 Our Auction Protocols

This section focuses on our auction protocols. First, we introduce the protocol PRIVATE BIDDING. It is also the protocol for the millionaires' problem. On the basis of PRIVATE BIDDING, we develop the auction protocol AUCTION1. It allows more than two players to compare their bids and finds the highest one. In this protocol, each player does not have any information about the value of the other players' bids, but he can know whose bid is higher than his, and whose bid is lower. We develop another auction protocol AUCTION2 to improve the security. In the protocol AUCTION2, each player knows only his own order among all. Finally, we give security analysis

of all our protocols.

2.1 Two-party Private Bidding (Millionaires' Problem)

We introduce the protocol for computing the greater-than function : $GT(x_1, x_2) = (x_1 > x_2)$ where $(x_1 > x_2) = 1$ if and only if $x_1 > x_2$. Yao's millionaires' problem [Yao82][Yao86] and Cachin's private bidding problem [Cac99] are both based on the secure computation of the greater-than function. In the millionaires' problem, two millionaires want to compare their fortunes, but would not like to reveal their assets. In the problem of private bidding, Alice wants to buy some goods from Bob. The deal will take place if and only if the price Alice offers is greater than Bob's selling price. We unify this kind of problems to the problem of comparison of two ℓ-bit strings, x_1, x_2, where $x_1 = x_{1,\ell}x_{1,\ell-1}\ldots x_{1,1}$, $x_2 = x_{2,\ell}x_{2,\ell-1}\ldots x_{2,1}$, which are possessed by two players, P_1 and P_2. Let X_i^1, X_i^0 be two sets of prefix strings such that

$$X_i^1 = \{x_{i,\ell}x_{i,\ell-1}\ldots x_{i,j+1} \mid x_{i,j} = 1, 1 \le j \le \ell\}$$
$$X_i^0 = \{x_{i,\ell}x_{i,\ell-1}\ldots x_{i,j+1} \mid x_{i,j} = 0, 1 \le j \le \ell\}$$

where we define the prefix string of $x_{i,\ell}$ as $x_{i,\ell+1} = \lambda$.
For example, let $x_1 = 234$. Convert x_1 to binary strings, $x_1 = 11101010$. Then we get

$$X_1^1 = \{\lambda, 1, 11, 1110, 111010\}, \text{ and}$$
$$X_1^0 = \{111, 11101, 1110101\}.$$

With two sets of prefix strings, we can compute the greater-than function by observing whether X_1^1 and X_2^0 have intersection, that is,

$$GT(x_1, x_2) = \begin{cases} 1 & \text{if } X_1^1 \cap X_2^0 \neq \emptyset \\ 0 & \text{if } X_1^1 \cap X_2^0 = \emptyset \end{cases}$$

For example, let $x_1 = 234$, $x_2 = 228$. Then

$$x_1 = 11101010$$
$$x_2 = 11100100$$
$$X_1^1 = \{\lambda, 1, 11, 1110, 111010\}$$
$$X_2^0 = \{111, 1110, 111001, 1110010\}$$

We can easily see that $X_1^1 \cap X_2^0 = \{1110\}$, so x_1 is greater than x_2. Up to now, we have introduced an idea of comparison between x_1 and x_2. In order to achieve the goal of secure computation, one player can not directly give prefix strings to the other party, otherwise it will disclose its secret. We use the technique of Diffie-Hellman key exchange [DH76] to commit each element of the prefix-string sets. As the scheme of the key exchange, there is a public input, a prime $p = 2q + 1$ where q is also a prime. Our computation

is all over the order-q subgroup G_q. At first, each player P_i prepares two prefix-string sets, X_i^1 and X_i^0, from his secret x_i and maps each binary prefix string into a value in G_q with a pre-specified hash function H. This process forms two ℓ-element sets, $A_i = \{a_{i,1}, a_{i,2} \dots a_{i,\ell}\}$ and $B_i = \{b_{i,1}, b_{i,2} \dots b_{i,\ell}\}$ where each element in A_i (resp. B_i) is a hash value from each element in X_i^1 (resp. X_i^0). If there are not enough elements, randomly choose values from G_q. The elements in A_i and B_i can be seen as the generators in the protocol of Diffie-Hellman key exchange. P_i securely checks whether A_i and B_j have intersection by checking if the key generated from $a_{i,s}$ and the key generated from $b_{j,t}$ are the same or not. The process is as follows. Each player P_i randomly chooses a variable u_i in Z_q^* to commit each element $a_{i,s}$ in A_i such that $m_{i,s} = a_{i,s}^{u_i}$, and randomly chooses a variable v_i in Z_q^* to commit each element $b_{i,s}$ in B_i such that $\mu_{i,s} = b_{i,s}^{v_i}$. He submits all values $m_{i,s}, \mu_{i,s}$ to the other one. Next, each party P_i computes $\alpha_{i,s} = m_{j,s}^{v_i}$ where $m_{j,s}$ is gotten from the other party, and transmits all values $\alpha_{i,s}$ to P_j. In the final, P_i computes $\beta_{i,s} = \mu_{j,s}^{u_i}$. With the information of $\alpha_{j,s} = a_{i,s}^{u_i v_j}$ and $\beta_{i,t} = b_{j,t}^{v_j u_i}$, P_i determines himself as the winner if there exists some elements such that $\alpha_{j,s} = \beta_{i,t}$. In other words, P_i is a winner if and only if A_i and B_j have intersection. The protocol is shown in Figure 1 PRIVATE BIDDING. We omit the modulo computation during the description of the protocol. For the reason of security, we use a permutation to mix the data sent to the other side.

2.2 Auction

We introduce the protocol for the first-price auction. The auction protocol is an extension of the two-party private bidding protocol. There are n bidders, $P_1, P_2, \dots, P_n$. Each of them has his ℓ-bit bidding price x_i. After the execution of the protocol, we want to find the highest bidding price and the player who bids the winning price. No trusted servers are needed in our protocol. Instead, a public bulletin board is used for communicating between bidders. Like the two-party private bidding protocol, there are two rounds of communication in this protocol. During each round, each bidder reads some information from the bulletin board and posts some messages on the board. After the communication, everyone checks if he is the winner or not. If the player finds his price the highest one, he opens his winning bid and posts some secrets to prove it. The scheme is similar with the protocol PRIVATE BIDDING in Figure 1. There is a public input, a prime $p = 2q+1$, and all computation is in G_q. Each player P_i prepares two sets, $A_i = \{a_{i,1}, a_{i,2}, \dots, a_{i,\ell}\}$ and $B_i = \{b_{i,1}, b_{i,2}, \dots, b_{i,\ell}\}$, in which the elements are computed from the elements in the prefix-string sets X_i^1 and X_i^0 with a hash function H. P_i chooses random variables u_i and v_i to commit the elements in A_i and B_i such that $m_{i,s} = a_{i,s}^{u_i}$ and $\mu_{i,s} = b_{i,s}^{v_i}$. Each player posts the values $m_{i,s}$ and

Public Input :

- (p, q) : p, q are random primes, $p = 2q + 1$ and q is k-bit where k is a security parameter.
- H : H is a hash function mapping the binary string to the value in G_q, which is the order-q subgroup of Z_p^*.

Secret Input of $P_i, i = 1, 2$:

- x_i : x_i P_i's secret.
- π_i : π_i is a permutation over $\{1, 2, \ldots, \ell\}$ for mix.

Protocol :

- **Preparation** : Each player P_i converts his secret x_i into two sets of prefix strings, X_i^1, X_i^0, and then computes two ℓ-element sets $A_i = \{a_{i,s} \mid a_{i,s} = H(x_{i,s}^1), x_{i,s}^1 \in X_i^1, 1 \leq s \leq \ell\}$ and $B_i = \{b_{i,s} \mid b_{i,s} = H(x_{i,s}^0), x_{i,s}^0 \in X_i^0, 1 \leq s \leq \ell\}$,$a_{i,s}, b_{i,s} \in G_q$. If there are not enough elements, randomly choose values from G_q.
- **First Round** :
  - P_1 : Choose a random variable $u_1 \in Z_q^*$, and compute $m_{1,s} = a_{1,\pi_1(s)}^{u_1}$ for $s = 1, 2, \ldots, \ell$.
  - P_2 : Choose a random variable $u_2 \in Z_q^*$, and compute $m_{2,s} = a_{2,\pi_2(s)}^{u_2}$ for $s = 1, 2, \ldots, \ell$.
  - $P_1 \longrightarrow P_2$: $m_{1,\ell}, m_{1,\ell-1}, \ldots, m_{1,1}$
  - $P_1 \longleftarrow P_2$: $m_{2,\ell}, m_{2,\ell-1}, \ldots, m_{2,1}$
- **Second Round** :
  - P_1 :
    1. Choose a random variable $v_1 \in Z_q^*$, and compute $\mu_{1,s} = b_{1,\pi_1(s)}^{v_1}$ for $s = 1, 2, \ldots, \ell$.
    2. Compute $\alpha_{1,s} = m_{2,\pi_1(s)}^{v_1}$ for $s = 1, 2, \ldots, \ell$.
  - P_2 :
    1. Choose a random variable $v_2 \in Z_q^*$, and compute $\mu_{2,s} = b_{2,\pi_2(s)}^{v_2}$ for $s = 1, 2, \ldots, \ell$.
    2. Compute $\alpha_{2,s} = m_{1,\pi_2(s)}^{v_2}$ for $s = 1, 2, \ldots, \ell$.
  - $P_1 \longrightarrow P_2$: $(\mu_{1,\ell}, \mu_{1,\ell-1}, \ldots, \mu_{1,1})$, $(\alpha_{1,\ell}, \alpha_{1,\ell-1}, \ldots, \alpha_{1,1})$
  - $P_1 \longleftarrow P_2$: $(\mu_{2,\ell}, \mu_{2,\ell-1}, \ldots, \mu_{2,1})$, $(\alpha_{2,\ell}, \alpha_{2,\ell-1}, \ldots, \alpha_{2,1})$
- **Final** :
  - P_1 : Compute $\beta_{1,s} = \mu_{2,s}^{u_1}$, for $s = 1, 2, \ldots, \ell$. Check if there exists some elements such that $\alpha_{2,s} = \beta_{1,t}$, $1 \leq s, t \leq \ell$. If it does, he knows x_1 is greater than x_2.
  - P_2 : Compute $\beta_{2,s} = \mu_{1,s}^{u_2}$, for $s = 1, 2, \ldots, \ell$. Check if there exists some elements such that $\alpha_{1,s} = \beta_{2,t}$, $1 \leq s, t \leq \ell$. If it does, he knows x_2 is greater than x_1.

Figure 1: PRIVATE BIDDING

$\mu_{i,s}$ on the bulletin board. Like the protocol PRIVATE BIDDING, each one computes $\alpha_{i,j,s} = m_{j,s}^{v_i}$ for all other players P_j and posts the values $\alpha_{i,j,s}$ on the board. To compare with P_j's bidding price, the player P_i reads $\mu_{j,s}$ and $\alpha_{j,i,s}$ from the board and computes $\beta_{i,j,s} = \mu_{j,s}^{u_i}$. P_i checks if there exists some values $\alpha_{j,i,s} = a_{i,s}^{u_i v_j}$ and $\beta_{i,j,t} = b_{j,t}^{v_j u_i}$ such that $\alpha_{j,i,s} = \beta_{i,j,t}$. If it does, he announces himself as the winner. The protocol is shown in Figure 2 AUCTION1. Permutations are also used for mixing the data posted on the board.

We must point out that AUCTION1 has weaker security. Although each bidding price is not revealed exactly, each bidder knows whose bids are higher than his and whose bids are lower than his after the communication. We modify the scheme to enhance the security by presenting another auction protocol AUCTION2, for which each bidder knows only his own order among all. The basic idea is that $x_i > x_j$ if and only if X_i^1 and X_j^0 have intersection. In particular, exactly one element appears in the intersection. Let $\Theta = \bigcup_{j \neq i} X_j^0$ [*]. If x_i is greater than all other x_j where $j \neq i$. X_i^1 and Θ must have intersection and there are $(n-1)$ elements in the intersection. The difference of this protocol is that each player P_i does the union for P_{i-1}. The public inputs, secret inputs of each player, and the preparation, first round, second round in this protocol are similar with those in the protocol AUCTION1. In order to conceal the comparing information $\alpha_{i,j,s}$, each player P_i chooses a key pair of his encryption scheme (E_i,D_i), publishes the encryption function E_i and keeps the decryption function D_i secret. A prime p is the public input where $p = 2q + 1$ and q is also a prime. All computation is in the order-q subgroup G_q. Each player prepares two sets $A_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,\ell}\}$ and $B_i = \{b_{i,1}, b_{i,2}, \ldots, b_{i,\ell}\}$ from prefix-string sets X_i^1 and X_i^0 with a hash function H. Random variables u_i and v_i are chosen by P_i for committing each element in A_i and B_i. P_i calculates $m_i = a_{i,s}^{u_i}$, $\mu_{i,s} = b_{i,s}^{v_i}$, and posts all values $m_{i,s}$, $\mu_{i,s}$ on the board. In the next round, P_i reads the data $m_{j,s}$, computes $\alpha_{i,j,s} = m_{j,s}^{v_i}$. In order to keep P_j from knowing the information of $\alpha_{i,j,s}$, P_i encrypts $\alpha_{i,j,s}$ by the encryption function E_{j+1} such that $e_{i,j,s} = E_{j+1}(\alpha_{i,j,s})$, and then submits $e_{i,j,s}$ to the board. In the third round, each player P_i reads all values $e_{j,i-1,s}$, and decrypts them by his decryption function D_i such that $\alpha_{j,i-1,s} = D_i(e_{j,i-1,s})$. P_i randomly chooses a variable w_i in Z_q^*, and computes two sets $\Gamma_{i-1} = \{\gamma_{i-1,1}, \gamma_{i-1,2}, \ldots, \gamma_{i-1,(n-1)\ell}\}$ and $\Delta_{i-1} = \{\delta_{i-1,1}, \delta_{i-1,2}, \ldots, \delta_{i-1,(n-1)\ell}\}$ for the player P_{i-1} [†] where each element in Γ_{i-1} (resp. Δ_{i-1}) is the value $\alpha_{j,i-1,s}$ (resp. $\mu_{j,s}$), $j \neq (i-1)$, to the power of w_i. Γ_{i-1} and Δ_{i-1} are posted on the board by P_i. In the final, P_i computes the set $\Delta_i' = \{\delta_{i,1}', \delta_{i,2}', \ldots, \delta_{i,(n-1)\ell}'\}$ where $\delta_{i,s}' = \delta_{i,s}^{u_i}$. He checks if there are $(n-1)$ elements in the intersection of Γ_i and Δ_i'. If it does, P_i announces himself as the winner. The protocol AUCTION2 is shown in Figure 3. All

[*] Θ may be a multi-set which allows duplicate items in one set.

[†] If $i = 1$, $i - 1$ means n.

Public Input :

- (p, q) : p, q are random primes, $p = 2q + 1$ and q is k-bit where k is a security parameter.
- H : H is a hash function mapping the binary string to the value in G_q, which is the order-q subgroup of Z_p^*.

Secret Input of P_i **for** $i = 1, 2, \ldots, n$:

- x_i : x_i is the bidding price.
- π_i : π_i is a permutation over $\{1, 2, \ldots, \ell\}$ for mix.

Protocol :

- **Preparation** : Each player P_i converts his bid x_i into two sets of prefix strings, X_i^1, X_i^0, and then computes two ℓ-element sets $A_i = \{a_{i,s} \mid a_{i,s} = H(x_{i,s}^1), x_{i,s}^1 \in X_i^1, 1 \leq s \leq \ell\}$ and $B_i = \{b_{i,s} \mid b_{i,s} = H(x_{i,s}^0), x_{i,s}^0 \in X_i^0, 1 \leq s \leq \ell\}$,$a_{i,s}, b_{i,s} \in G_q$. If there are not enough elements, randomly choose values from G_q.
- **First Round** : Each player P_i does the following :
  1. Choose a random variable $u_i \in Z_q^*$, and compute $m_{i,s} = a_{i,\pi_i(s)}^{u_i}$ for $s = 1, 2, \ldots, \ell$.
  2. Post the values $m_{i,s}$ on the bulletin board for $s = 1, 2, \ldots, \ell$.
- **Second Round** : Each player P_i does the following :
  1. Choose a random variable $v_i \in Z_q^*$, and compute $\mu_{i,s} = b_{i,\pi_i(s)}^{v_i}$ for $s = 1, 2, \ldots, \ell$.
  2. In accordance with the variables $m_{j,s}$ that P_j, $1 \leq j \neq i \leq n$, posts, compute $\alpha_{i,j,s} = m_{j,\pi_i(s)}^{v_i}$ for $s = 1, 2, \ldots, \ell$.
  3. Post the values $\mu_{i,s}$ and $\alpha_{i,j,s}$ on the board where $1 \leq s \leq \ell$, $1 \leq j \neq i \leq n$.
- **Final** : Each player P_i does the following :
  - For comparing with P_j, $1 \leq j \neq i \leq n$, compute $\beta_{i,j,s} = \mu_{j,s}^{u_i}$ for $s = 1, 2, \ldots, \ell$. Check if there exists some elements such that $\alpha_{j,i,s} = \beta_{i,j,t}$, $1 \leq s, t \leq \ell$. If it does, he knows x_i is greater than x_j.
  - After all checks, if x_i is greater than all other x_j, $1 \leq j \neq i \leq n$, post the variables u_i,v_i and his winning bid x_i. Other bidders can verify it by the same way.

Figure 2: AUCTION1

posted values are also disordered by permutations.

2.3 Security Analysis

We discuss the security requirements of our protocols. A secure private bidding or auction protocol must satisfy two requirements : correctness and privacy. In the proof of correctness, we show that the player announces that he is the winner if and only if he has the highest bidding price. In the proof of privacy, we require that nobody gets the others' bidding prices except the winning price. As we mentioned above, after the execution of the protocol AUCTION1, each player knows the greater-than or less-than relationship between his bid and those of the other bidders. However, he learns no information about the relationship between any pair of the others. After the execution of the protocol AUCTION2, each bidder knows his order among all, but no information about others' orders.

2.3.1 Correctness

The correctness of our three protocols is basically based on the correctness of the greater-than function we present. Given two ℓ-bit secrets, x_1 and x_2, x_1 is greater than x_2 if and only if the intersection of X_1^1 and X_2^0 is not empty and there is only one element in the intersection. The error occurs when players randomly choose some variables in G_q to fill two sets A_i and B_j such that X_i^1 and X_j^0 have no intersection but A_i and B_j have intersection. The probability we randomly select the same element in G_q is $O(2^{-k})$. If k is large enough (ex. $k = 1024$), the error probability is negligible. We can disregard it. Besides, in the protocol AUCTION2, the error also occurs when $\gamma_{i,s}(= a_{i,s}^{u_i v_j w_{i+1}})$ equals $\delta'_{i,t}(= b_{h,t}^{u_i v_h w_{i+1}})$ but $a_{i,s} \neq b_{h,t}$, $j \neq h$. The probability is $O(2^{-k})$. We can also ignore it. Therefore, after the protocols PRIVATE BIDDING, AUCTION1, AUCTION2, the winner is the player who bids the highest price with overwhelming probability.

2.3.2 Privacy

An important goal of the private bidding or sealed-bid auctions is to keep each player's bidding price secret during the protocol. We say that one player P_i gets no information about any other's bidding price x_j if the view of P_i in the execution of the protocol where P_j bids x_j is indistinguishable from the view in the execution of the protocol where P_j bids x_j^*.

Based on the DDH assumption, the following pairs of ensembles can be reduced to be polynomially indistinguishable.

- $R^* = \{R_n^*\}$ and $D^* = \{D_n^*\}$:
  - $R_n^* = (p, q, g_1, g_2, g_1^a, g_2^b)$

Public Input :

- (p, q) : p, q are random primes, $p = 2q + 1$ and q is k-bit where k is a security parameter.
- H : H is a hash function mapping the binary string to the value in G_q, which is the order-q subgroup of Z_p^*.
- $E_i, 1 \leq i \leq n$: E_i is the encryption function of the bidder P_i.

Secret Input of P_i :

- x_i : x_i is the bidding price.
- π_i : π_i is a permutation over $\{1, 2, \ldots, \ell\}$ for mix.
- D_i : D_i is the decryption function for E_i.

Protocol :

- **Preparation** : Each player P_i converts his bid x_i into two sets of prefix strings, X_i^1, X_i^0, and then computes two ℓ-element sets $A_i = \{a_{i,s} \mid a_{i,s} = H(x_{i,s}^1), x_{i,s}^1 \in X_i^1, 1 \leq s \leq \ell\}$ and $B_i = \{b_{i,s} \mid b_{i,s} = H(x_{i,s}^0), x_{i,s}^0 \in X_i^0, 1 \leq s \leq \ell\}$,$a_{i,s}, b_{i,s} \in G_q$. If there are not enough elements, randomly choose values from G_q.
- **First Round** : Each player P_i does the following :
  1. Choose a random variable $u_i \in Z_q^*$, and compute $m_{i,s} = a_{i,\pi_i(s)}^{u_i}$ for $s = 1, 2, \ldots, \ell$.
  2. Post the values $m_{i,s}$ on the bulletin board for $s = 1, 2, \ldots, \ell$.
- **Second Round** : Each player P_i does the following :
  1. Choose a random variable $v_i \in Z_q^*$, and compute $\mu_{i,s} = b_{i,\pi_i(s)}^{v_i}$ for $s = 1, 2, \ldots, \ell$.
  2. In accordance with the variables $m_{j,s}$ that P_j, $1 \leq j \neq i \leq n$ posts, compute $\alpha_{i,j,s} = m_{j,\pi_i(s)}^{v_i}$, and use the encryption function E_{j+1} to encrypt $\alpha_{i,j,s}$ such that $e_{i,j,s} = E_{j+1}(\alpha_{i,j,s})$ for $s = 1, 2, \ldots, \ell$.
  3. Post the values $\mu_{i,s}$ and $e_{i,j,s}$ on the board where $1 \leq s \leq \ell$, $1 \leq j \neq i \leq n$.
- **Third Round** : Each player P_i does the following :
  1. Use the decryption function D_i to decrypt all values $e_{j,i-1,s}$ such that $\alpha_{j,i-1,s} = D_i(e_{j,i-1,s})$ where $1 \leq j \neq (i-1) \leq n$, $1 \leq s \leq \ell$.
  2. Choose a random variable $w_i \in Z_q^*$, and compute two sets $\Gamma_{(i-1)} = \{\alpha_{j,i-1,s}^{w_i} | 1 \leq j \neq (i-1) \leq n, 1 \leq s \leq \ell\}$ and $\Delta_{(i-1)} = \{\mu_{j,s}^{w_i} | 1 \leq j \neq (i-1) \leq n, 1 \leq s \leq \ell\}$. Use the mix technique to disorder the elements in each set.
  3. Post the sets $\Gamma_{(i-1)} = \{\gamma_{(i-1),1}, ..., \gamma_{(i-1),(n-1)\ell}\}$ and $\Delta_{(i-1)} = \{\delta_{(i-1),1}, ..., \delta_{(i-1),(n-1)\ell}\}$ on the bulletin board.
- **Final** : Each player P_i does the following :
  - Read the information of two sets Γ_i and Δ_i on the board. Then compute $\Delta_i' = \{\delta_{i,s}' | \delta_{i,s}' = \delta_{i,s}^{u_i}, 1 \leq s \leq (n-1)\ell\}$.
  - Check how many pairs $(\gamma_{i,s}, \delta_{i,t}')$ there exists such that $\gamma_{i,s} = \delta_{i,t}'$, $1 \leq s, t \leq \ell$. If there are (n-1) or more equivalent pairs, he is the winner. Then post the variables u_i,v_i and the wining price x_i. Other bidders can verify it by the same way.

Figure 3: AUCTION2

- $D_n^* = (p, q, g_1, g_2, g_1^a, g_2^a)$

- $S = \{S_n\}$ and $T = \{T_n\}$
  - $S_n = (p, q, g_1, g_2, h_1, h_2, g_1^a, g_2^a)$
  - $T_n = (p, q, g_1, g_2, h_1, h_2, h_1^a, h_2^a)$

- $S' = \{S'_n\}$ and $T' = \{T'_n\}$
  - $S'_n = (p, q, g_1, g_2, ..., g_\ell, h_1, h_2, ..., h_\ell, g_1^a, g_2^a, ..., g_\ell^a)$
  - $T'_n = (p, q, g_1, g_2, ..., g_\ell, h_1, h_2, ..., h_\ell, h_1^a, h_2^a, ..., h_\ell^a)$

where p is an n-bit prime, $p = 2q + 1$, q is also a prime, g_i,h_i are generators of the order-q subgroup G_q of Z_p^*, and a, b are chosen uniformly from Z_q^*.

In the protocols PRIVATE BIDDING and AUCTION1, we say P_i gets no information about P_j's bidding price if the following two views are polynomially indistinguishable :

- $\langle (m_{j,\ell}, ..., m_{j,1}), (\mu_{j,\ell}, ..., \mu_{j,1}), (\alpha_{j,i,\ell}, ..., \alpha_{j,i,1}) \rangle$ with P_j's bid x_j
- $\langle (m^*_{j,\ell}, ..., m^*_{j,1}), (\mu^*_{j,\ell}, ..., \mu^*_{j,1}), (\alpha^*_{j,i,\ell}, ..., \alpha^*_{j,i,1}) \rangle$ with P_j's bid x^*_j

where x_j and x^*_j satisfy the result, ex. $x_j < x_i$ and $x^*_j < x_i$. Based on the polynomial indistinguishability of the two ensembles S' and T', the following pairs of views are polynomially indistinguishable :

1.
   - $\langle p, q, a_{j,\ell}, ..., a_{j,1}, a^*_{j,\ell}, ..., a^*_{j,1}, m_{j,\ell} = (a_{j,\ell})^{u_j}, ..., m_{j,1} = (a_{j,1})^{u_j} \rangle$
   - $\langle p, q, a_{j,\ell}, ..., a_{j,1}, a^*_{j,\ell}, ..., a^*_{j,1}, m^*_{j,\ell} = (a^*_{j,\ell})^{u_j}, ..., m^*_{j,1} = (a^*_{j,1})^{u_j} \rangle$
2.
   - $\langle p, q, b_{j,\ell}, ..., b_{j,1}, b^*_{j,\ell}, ..., b^*_{j,1}, \mu_{j,\ell} = (b_{j,\ell})^{v_j}, ..., \mu_{j,1} = (b_{j,1})^{v_j} \rangle$
   - $\langle p, q, b_{j,\ell}, ..., b_{j,1}, b^*_{j,\ell}, ..., b^*_{j,1}, \mu^*_{j,\ell} = (b^*_{j,\ell})^{v_j}, ..., \mu^*_{j,1} = (b^*_{j,1})^{v_j} \rangle$

The two views $(\alpha_{j,\ell}, ..., \alpha_{j,1})$ and $(\alpha^*_{j,\ell}, ..., \alpha^*_{j,1})$ are identically distributed since $\alpha_{j,s} = a_{i,s}^{u_i v_j} = \alpha^*_{j,s}$. Therefore, two views of P_i are polynomially indistinguishable. No player gets any information about any other's bidding price. We must point out that when x_i is greater than x_j, P_i knows that there exists a bit b such that $b = 1$ in x_i and $b = 0$ in x_j, and the prefix strings of b in x_i and in x_j are equal. However, if there are more than one bit 1 in x_i, P_i still gets no information about which bit 1 makes x_i greater than x_j since each player uses a permutation to randomize the sequence of the variables he sends to the other party.

The protocol AUCTION2 is a modification of the protocol AUCTION1, but the security is stronger such that no bidder gets any information about the others' bidding prices except the order of his bid among all. We say that the player P_i has no information about the others' bids x_j except

his order ω if given two $(n-1)$-bid vectors $\langle x_1, ..., x_{i-1}, x_{i+1}, ..., x_n\rangle$ and $\langle x_1^*, ..., x_{i-1}^*, x_{i+1}^*, ..., x_n^*\rangle$ such that $(\omega - 1)$ bids x_j (resp. x_j^*) are greater than x_i and others are less, P_i can not distinguish the following two views :

- $\langle\{m_{j,s}, \mu_{j,s}, e_{j,i,s}, \gamma_{i,t}, \delta_{i,t} | 1 \le j \ne i \le n, 1 \le s \le \ell, 1 \le t \le (n-1)\ell\}\rangle$ with input $\langle x_1, ..., x_{i-1}, x_{i+1}, ..., x_n\rangle$
- $\langle\{m_{j,s}^*, \mu_{j,s}^*, e_{j,i,s}^*, \gamma_{i,t}^*, \delta_{i,t}^* | 1 \le j \ne i \le n, 1 \le s \le \ell, 1 \le t \le (n-1)\ell\}\rangle$ with input $\langle x_1^*, ..., x_{i-1}^*, x_{i+1}^*, ..., x_n^*\rangle$

The proof is similar with that of the protocol AUCTION1, it can be shown that the following pairs of views are polynomially indistinguishable.

1. - $\langle p, q, \{a_{j,s}, a_{j,s}^*, m_{j,s} = (a_{j,s})^{u_j} | 1 \le j \ne i \le n, 1 \le s \le \ell\}\rangle$
   - $\langle p, q, \{a_{j,s}, a_{j,s}^*, m_{j,s}^* = (a_{j,s}^*)^{u_j} | 1 \le j \ne i \le n, 1 \le s \le \ell\}\rangle$
2. - $\langle p, q, \{b_{j,s}, b_{j,s}^*, \mu_{j,s} = (b_{j,s})^{v_j} | 1 \le j \ne i \le n, 1 \le s \le \ell\}\rangle$
   - $\langle p, q, \{b_{j,s}, b_{j,s}^*, \mu_{j,s}^* = (b_{j,s}^*)^{v_j} | 1 \le j \ne i \le n, 1 \le s \le \ell\}\rangle$
3. - $\langle p, q, \{b_{j,s}, b_{j,s}^*, \delta_{i,t} = (b_{j,s})^{v_j w_{i+1}} |$ $1 \le j \ne i \le n, 1 \le s \le \ell, 1 \le t \le (n-1)\ell\}\rangle$
   - $\langle p, q, \{b_{j,s}, b_{j,s}^*, \delta_{i,t}^* = (b_{j,s}^*)^{v_j w_{i+1}} |$ $1 \le j \ne i \le n, 1 \le s \le \ell, 1 \le t \le (n-1)\ell\}\rangle$

Since $e_{j,i,s} = E_{i+1}(\alpha_{j,i,s}) = E_{i+1}((a_{i,s})^{u_i v_j})$, $e_{j,i,s}^* = E_{i+1}(\alpha_{j,i,s}^*) = E_{i+1}((a_{i,s})^{u_i v_j})$ and $\gamma_{i,s} = (a_{i,s})^{u_i v_j w_{i+1}} = \gamma_{i,s}^*$, the following pairs of views are identically distributed

1. - $\langle\{e_{j,i,s} | 1 \le j \ne i \le n, 1 \le s \le \ell\}\rangle$
   - $\langle\{e_{j,i,s}^* | 1 \le j \ne i \le n, 1 \le s \le \ell\}\rangle$
2. - $\langle\{\gamma_{i,t} | 1 \le t \le (n-1)\ell\}\rangle$
   - $\langle\{\gamma_{i,t}^* | 1 \le t \le (n-1)\ell\}\rangle$

Thus, two views of P_i are polynomially indistinguishable. Since x_j (resp. x_j^*) is chosen to satisfy the result of the protocol, P_i finds the order of his bid ω in the final phase. Therefore, no player gets any information about the others' bids except the order of his bid among all.

2.4 Performance

We discuss the computation and communication cost of our protocols. Let n be the number of bidders, ℓ be the bit length of bidding prices, and p be the modulo of the discrete logarithm problem. The protocol PRIVATE BIDDING in Figure 1 requires two rounds of communication. The amount of data in each

round is $O(\ell)$. Each bidder spends about 4ℓ modular exponentiations in Z_p. The protocol AUCTION1 in Figure 2 requires two rounds of communication. In the first round, the amount of data on the bulletin board is $n\ell$. In the second round, the amount is $n^2\ell$. Each bidder spends about $2n\ell$ modular exponentiations in Z_p. The protocol AUCTION2 in Figure 3 requires three rounds of communication. In the first round, the amount of data on the board is $n\ell$. In the second round, the amount is $n^2\ell$. In the third round, the amount is $2n^2\ell - 2n\ell$. Each bidder spends about $4n\ell - 2\ell$ modular exponentiations in Z_p.

We compare our protocols with the existing private bidding and sealed-bid auction protocols. Table 1 shows the types of trusted third parties used in all the protocols compared. Table 2 shows the computation complexity. Table 3 shows the communication complexity. Table 4 shows the privacy of auction protocols.

Table 1: comparison of trusted third parties

protocol	**trusted third party**
[Cac99] private bidding	1 oblivious server
[Cac99] auction	2 non-colluding servers
[Fis] private bidding	1 oblivious server
[BS] auction	1 semi-trusted server
our PRIVATE BIDDING	none
our AUCTION1	1 public bulletin board
our AUCTION2	1 public bulletin board

Table 2: comparison of computation complexity

	computation complexity	
protocol	**bidder**	**server**
[Cac99] private bidding	ℓ encryptions	2ℓ modular exponentiations
[Cac99] auction	$2n\ell$ encryptions	$2n\ell$ modular exponentiations
[Fis] private bidding	$6\ell\lambda + \ell$ modular multiplications	$2\ell\lambda$ decryptions
[BS] auction	$n\ell$ encryptions	$n(n-2)\ell^{(n-1)}$ modular multiplications
our PRIVATE BIDDING	4ℓ modular exponentiations	
our AUCTION1	$2n\ell$ modular exponentiations	
our AUCTION2	$4n\ell - 2\ell$ modular exponentiations	

Table 3: comparison of communication complexity

protocol	**round(s)**		
	bidder ⇔ bidder	**bidder ⇔ server(board)**	**server ⇔ server**
[Cac99] bidding	1	1	
[Cac99] auction		1	$O(n)$
[Fis] bidding	1	1	
[BS] auction		1	
our PRIVATE BIDDING	2		
our AUCTION1		2	
our AUCTION2		3	

protocol	**message complexity (in blocks)**			
	bidder ⇔ bidder	**bidder ⇔ server (bidder ⇔ board)**	**server ⇔ server**	**total amount on the board**
[Cac99] bidding		ℓ		
[Cac99] auction		$2n\ell$	ℓ	
[Fis] bidding	ℓ	$\ell\lambda$		
[BS] auction		$n\ell$		$n\ell^{n-1}$
our PRIVATE BIDDING	2ℓ			
our AUCTION1		$n\ell$		$n^2\ell$
our AUCTION2		$2n\ell - 2\ell$		$2n^2\ell - 2n\ell$

Table 4: comparison of privacy

protocol	**privacy**
[Cac99] auction	reveal the greater-than and less-than relationship of bids to one auctioneer
[BS] auction	reveal nothing
our AUCTION1	reveal the greater-than and less-than relationship of bids to bidders
our AUCTION2	reveal the order of bids among all to bidders

3 Conclusion

In this paper, we proposed a novel construction for the secure computation of the greater-than function. We applied it to the two-party private bidding and first-price sealed-bid auction. Compared with existing schemes, our protocols require no trusted third parties. Instead, a public bulletin board is used. In the private bidding protocol PRIVATE BIDDING, it needs two rounds of communication between two bidders. In the auction protocol AUCTION1, it takes two rounds of communication between the bidders and the bulletin board, and in AUCTION2, it takes three rounds. All computation is linear in the bit length of the bidding price ℓ, the number of bidders n, and the security parameter k.

Finally, we want to point out some possible improvements or directions for further work.

- Enhanced Security : The auction protocols we developed reveal the partial order of bids to bidders.
- Robustness : An actively cheating player might collapse the process. For example, there are no good ways to prevent a bidder from preparing incorrect prefix-string sets of his bidding price. Although an adversary gets nothing by cheating, his improper play might cause a failure to the auction.
- Vickrey auction : The Vickrey acution has better revenue since the optimal strategy for each bidder is to bid his true value in this type of auctions. Some researchers are devoted to Vickrey or $M + 1$-st price auctions. However, their methods need the support of trusted third parties.

References

[AS02] Masayuki Abe and Koutarou Suzuki. M+1-st price auction using homomorphic encryption. *Public Key Cryptography 2002*, 2002.

[Boy00] Colin Boyd. Security issues for electronic auctions. Technical report, HP Labs 2000 Technical Report, 2000.

[BS] Olivier Baudron and Jacques Stern. Non-interactive private auctions. *Financial Cryptography 2001.*

[Cac99] Christian Cachin. Efficient private bidding and auctions with an oblivious third party. *The 6th ACM on Computer and Communications Security*, 1999.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory 22*, pages 644–654, 1976.

[Fis] Marc Fischlin. A cost-effective pay-per-multiplication comparison method for millionaires. *RSA 2001.*

[FR96] Matthew K. Franklin and Michael K. Reiter. The design and implementation of a secure auction service. *IEEE Transaction on Software Engineering*, 22(5):302–312, May 1996.

[HTK98] Michael Harkavy, J.D. Tygar, and Hiroaki Kikuchi. Electronic auctions with private bids. *The Third USENIX Workshop on Electronic Commerce*, pages 61–83, 1998.

[KF98] Manoj Kumar and Stuart I. Feldman. Internet auctions. *The Third USENIX Workshop on Electronic Commerce*, pages 49–60, 1998.

[Kik01] Hiroaki Kikuchi. (m+1)st-price auction. *The Fifth International Conference on Financial Cryptography '01*, pages 291–298, February 2001.

[MvOV96] Alfed J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of APPLIED CRYPTOGRAPHY.* CRC Press, October 1996.

[NPS99] Moni Naor, Benny Pinkas, and Reuben Summner. Privacy preserving auctions and mechanism design. *The First ACM Conference on Electronic Commerce*, November 1999.

[Sak00] Kazue Sako. An auction protocol which hides bids of losers. *Public Key Cryptography 2000*, pages 422–432, 2000.

[SS99] Stuart G. Stubblebine and Paul F. Syverson. Fair on-line auctions without special trusted parties. *Financial Cryptography '99*, pages 230–240, 1999.

[Yao82] A. Yao. Protocols for secure computation. *The 23rd IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[Yao86] A. Yao. How to generate and exchange secrets. *The 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.