

A Secure $M + 1$ st Price Auction Protocol based on Bit Slice Circuits

Takuho Mitsunaga *

Yoshifumi Manabe †

Tatsuaki Okamoto ‡

Abstract— This paper presents an efficient secure auction protocol for $M + 1$ st price auction. In our proposed protocol, bidding prices are represented as binary expression. Thus, if the bidding price is an integer up to p and the number of bidders is m , the complexity of our protocol is a polynomial of $\log p$ and m , while in previous secure $M + 1$ st price auction protocols, the complexity is a polynomial of p and m . We apply the Boneh-Goh-Nissim encryption to the mix-and-match protocol to reduce the computation costs.

Keywords: secure auction, $M+1$ st price auction, Boneh-Goh-Nissim encryption, mix-and-match protocol

1 Introduction

1.1 Background

Recently, as the Internet has expanded, many researchers have become interested in secure auction protocols and various schemes have been proposed to ensure the safe transaction of sealed-bid auctions. A secure auction is a protocol in which each player can find only the highest bid and its bidder (called the first price auction) or the second highest bid and the first price bidder (called the second price auction). There is also a generalized auction protocol called $M + 1$ st price auction. The $M + 1$ st price auction is a type of sealed-bid auction for selling M units of a single kind of goods, and the $M + 1$ st highest price is the winning price. M bidders who bid higher prices than the winning price are the winning bidders, and each winning bidder buys one unit of the goods at the $M + 1$ st winning price.

A simple solution is to assume a trusted auctioneer. Bidders encrypt their bids and send them to the auctioneer, and the auctioneer decrypts them to decide the winner. To remove the trusted auctioneer, some secure multi-party protocols have been proposed. The common essential idea is the use of threshold cryptosystems, where a private decryption key is shared by the players. Jakobsson and Juels proposed a secure MPC protocol to evaluate a function comprising a logical circuit, called mix-and-match [6]. As for a target function f and the circuit that calculates f , C_f , all players evaluate each gate in C_f based on their encrypted inputs and the evaluations of all the gates in turn lead to the evaluation of f . Based on the mix-and-match protocol, we can easily find a secure auction protocol by repeating the millionaires' problem for two players. Kurosawa and Ogata suggested the "bit-slice auction", which is an auction protocol that is more efficient than the one

based on the millionaire's problem [8].

Boneh, Goh and Nissim suggested a public evaluation system for 2-DNF formula based on an encryption of Boolean variables [3]. Their protocol is based on Paillier's scheme [13], so it has additive homomorphism in addition to the bilinear map, which allows one multiplication on encrypted values. As a result, this property allows the evaluation of multivariate polynomials with the total of degree two on encrypted values.

In this paper, we introduce an efficient secure auction protocol for $M + 1$ st price auction, in which if the bidding price is an integer up to p and the number of bidders is m , the complexity of our protocol is a polynomial of $\log p$ and m .

1.2 Related works

As related works, there are many secure auction protocols, however, they have problems such as those described hereafter. The secure auction scheme for first price auction proposed by Franklin and Reiter [5] does not provide full privacy, since at the end of an auction players can know the other players' bids. Naor, Pinkas and Sumner achieved a secure second price auction by combining Yao's secure computation with oblivious transfer assuming two types of auctioneers [10]. However, the cost of the bidder communication is high because it proceeds bit by bit using the oblivious transfer protocol. Juels and Szydlo improved the efficiency and security of this scheme with two types of auctioneers through verifiable proxy oblivious transfer [7], which still has a security problem in which if both types of auctioneers collaborate they can retrieve all bids. Mitsunaga, Manabe, Okamoto suggested secure auction protocols for first and second price auction. They applied Boneh-Goh-Nissim Encryption to bit-slice auction to improve an efficiency of calculation cost [11]. Lipmaa, Asokan and Niemi proposed an efficient $M + 1$ st secure auction scheme [9]. In their scheme, the trusted auction authority can know the bid statistics.

* Kobe Digital Labo / Kyoto University

† NTT Communication Science Laboratories

‡ NTT Information Sharing Platform Laboratories

Abe and Suzuki suggested a secure auction scheme for the $M + 1st$ auction based on homomorphic encryption [1]. However in their scheme, a player's bid is not a binary expression. So, its time complexity is $O(m2^k)$ for a m -player and k -bit bidding price auction. Tamura, Shiotsuki and Miyaji proposed an efficient proxy-auction for first price auction [15]. This scheme only considers the comparison between two sealed bids, the current highest bid and a new bid. However, this scheme does not consider multiple players because of the property of the proxy-auction.

1.3 Our result

This paper presents an efficient secure auction protocol for $M + 1st$ price auction. In our proposed protocol, bidding prices are represented as binary expression. Thus, if the bidding price is an integer up to p and the number of bidders is m , the complexity of our protocol is a polynomial of $\log p$ and m , while in previous secure $M + 1st$ price auction protocols, the complexity is a polynomial of p and m .

2 Preliminaries

2.1 The model of auctions and outline of auction protocols

This model involves n players, denoted by $P_1, P_2, \dots, P_n$ and assumes that there exists a public board. The players agree in advance on the presentation of the target function, f as a circuit C_f . The aim of the protocol is for players to compute $f(B_1, \dots, B_n)$ without revealing any additional information. Its outline is as follows.

1. **Input stage:** Each P_i ($1 \leq i \leq n$) computes ciphertexts of the bits of B_i and broadcasts them and proves that the ciphertext represents 0 or 1 by using the zero-knowledge proof technique in [3].
2. **Mix and Match stage:** The players blindly evaluates each gate, G_j , in order.
3. **Output stage:** After evaluating the last gate G_N , the players obtain O_N , a ciphertext encrypting $f(B_1, \dots, B_n)$. They jointly decrypt this ciphertext value to reveal the output of function f .

2.1.1 Requirements for the encryption function

Let E be a public-key probabilistic encryption function. We denote the set of encryptions for a plaintext m by $E(m)$ and a particular encryption of m by $c \in E(m)$.

Function E must satisfy the following properties.

1. Homomorphic property There exist polynomial time computable operations, $^{-1}$ and $\otimes$, as follows. For a large prime q ,

1. If $c \in E(m)$, then $c^{-1} \in E(-m \bmod q)$.

2. If $c_1 \in E(m_1)$ and $c_2 \in E(m_2)$, then $c_1 \otimes c_2 \in E(m_1 + m_2 \bmod q)$.

For a positive integer a , define

$$a \cdot e = \underbrace{c \otimes c \otimes \dots \otimes c}_a.$$

2. Random re-encryption Given $c \in E(m)$, there is a probabilistic re-encryption algorithm that outputs $c' \in E(m)$, where c' is uniformly distributed over $E(m)$.

3. Threshold decryption For a given ciphertext $c \in E(m)$, any t out of n players can decrypt c along with a zero-knowledge proof of the correctness. However, any $t-1$ out of n players cannot decrypt c .

2.1.2 MIX protocol

The MIX protocol [4] takes a list of ciphertexts, $(\xi_1, \dots, \xi_L)$, and outputs a permuted and re-encrypted list of the ciphertexts $(\xi'_1, \dots, \xi'_L)$ without revealing the relationship between $(\xi_1, \dots, \xi_L)$ and $(\xi'_1, \dots, \xi'_L)$, where ξ_i or ξ'_i can be a single ciphertext c , or a list of l ciphertexts, $(c_1, \dots, c_l)$, for some $l > 1$. For all players to verify the validity of $(\xi'_1, \dots, \xi'_L)$, we use the universal verifiable MIX net protocol described in [14].

2.1.3 Plaintext equality test

Given two ciphertexts $c_1 \in E(v_1)$ and $c_2 \in E(v_2)$, this protocol checks if $v_1 = v_2$. Let $c_0 = c_1 \otimes c_2^{-1}$.

1. (Step 1) For each player P_i (where $i = 1, \dots, n$):
 P_i chooses a random element $a_i \in \mathbb{Z}_q^*$ and computes $z_i = a_i \cdot c_0$. He broadcasts z_i and proves the validity of z_i in zero-knowledge.
2. (Step 2) Let $z = z_1 \otimes z_2 \otimes \dots \otimes z_n$. The players jointly decrypt z using threshold verifiable decryption and obtain plaintext v . Then it holds that

$$v = \begin{cases} 0 & \text{if } v_1 = v_2 \\ \text{random} & \text{otherwise} \end{cases}$$

2.1.4 Mix and Match Stage

For each logical gate, $G(x_1, x_2)$, of a given circuit, n players jointly computes $E(G(x_1, x_2))$ from $c_1 \in E(x_1)$ and $c_2 \in E(x_2)$ keeping x_1 and x_2 secret. For simplicity, we show the mix-and-match stage for AND gate.

1. n players first consider the standard encryption of each entry in the table shown below.
2. By applying a MIX protocol to the four rows of the table, n players jointly compute blinded and permuted rows of the table. Let the i th row be (a'_i, b'_i, c'_i) for $i = 1, \dots, 4$.
3. n players next jointly find the row i such that the plaintext of c_1 is equal to that of a'_i and the plaintext of c_2 is equal to that of b'_i by using the plaintext equality test protocol.

Table 1: Mix-and-match table for AND

x_1	x_2	$x_1 \wedge x_2$
$a'_1 \in E(0)$	$b'_1 \in E(0)$	$c'_1 \in E(0)$
$a'_2 \in E(0)$	$b'_2 \in E(1)$	$c'_2 \in E(0)$
$a'_3 \in E(1)$	$b'_3 \in E(0)$	$c'_3 \in E(0)$
$a'_4 \in E(1)$	$b'_4 \in E(1)$	$c'_4 \in E(1)$

4. For the row i , it holds that $c'_i \in E(x_1 \wedge x_2)$.

2.2 Evaluating 2-DNF formulas on ciphertexts

Given encrypted Boolean variables $x_1, \dots, x_n \in \{0, 1\}$, a mechanism for public evaluation of a 2-DNF formula was suggested in [3]. They presented a homomorphic public key encryption scheme based on finite groups of composite order that supports a bilinear map. In addition, the bilinear map allows for one multiplication on encrypted values. As a result, their system supports arbitrary additions and one multiplication on encrypted data. This property in turn allows the evaluation of multivariate polynomials of a total degree of two on encrypted values.

2.2.1 Bilinear groups

Their construction makes use of certain finite groups of composite order that supports a bilinear map. We use the following notation.

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two (multiplicative) cyclic groups of finite order n .
2. g is a generator of $\mathbb{G}$.
3. e is a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$.

2.2.2 Subgroup decision assumption

We define algorithm $\mathcal{G}$ such that given security parameter $\tau \in \mathbb{Z}^+$ outputs a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ where $\mathbb{G}, \mathbb{G}_1$ are groups of order $n = q_1 q_2$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ is a bilinear map. On input τ , algorithm $\mathcal{G}$ works as indicated below,

1. Generate two random τ -bit primes, q_1 and q_2 and set $n = q_1 q_2 \in \mathbb{Z}$.
2. Generate a bilinear group $\mathbb{G}$ of order n as described above. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ be the bilinear map.
3. Output $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$.

We note that the group action in $\mathbb{G}$ and $\mathbb{G}_1$ as well as the bilinear map can be computed in polynomial time.

Let $\tau \in \mathbb{Z}^+$ and let $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ be a tuple produced by $\mathcal{G}$ where $n = q_1 q_2$. Consider the following problem. Given $(n, \mathbb{G}, \mathbb{G}_1, e)$ and an element $x \in \mathbb{G}$, output '1' if the order of x is q_1 and output '0' otherwise, that is, without knowing the factorization of the group order n , decide if an element x is in a subgroup of $\mathbb{G}$. We refer to this problem as the subgroup decision problem.

2.2.3 Homomorphic public key system

We now describe the proposed public key system which resembles the Pallier [13] and the Okamoto-Uchiyama encryption schemes [12]. We describe the three algorithms comprising the system.

- 1.KeyGen** Given a security parameter $\tau \in \mathbb{Z}$, run $\mathcal{G}$ to obtain a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$. Let $n = q_1 q_2$. Select two random generators, g and $u \xleftarrow{R} \mathbb{G}$ and set $h = u^{q_2}$. Then h is a random generator of the subgroup of $\mathbb{G}$ of order q_1 . The public key is $PK = (n, \mathbb{G}, \mathbb{G}_1, e, g, h)$. The private key is $SK = q_1$.

- 2.Encrypt(PK, M)** We assume that the message space consists of integers in set $\{0, 1, \dots, T\}$ with $T < q_2$. We encrypt the binary representation of bids in our main application, in the case $T = 1$. To encrypt a message m using public key PK , select a random number $r \in \{0, 1, \dots, n-1\}$ and compute

$$C = g^m h^r \in \mathbb{G}.$$

Output C as the ciphertext.

- 3.Decrypt(SK, C)** To decrypt a ciphertext C using the private key $SK = q_1$, observe that $C^{q_1} = (g^m h^r)^{q_1} = (g^{q_1})^m$. Let $\hat{g} = g^{q_1}$. To recover m , it suffices to compute the discrete log of C^{q_1} base $\hat{g}$.

2.2.4 Homomorphic properties

The system is clearly additively homomorphic. Let $(n, \mathbb{G}, \mathbb{G}_1, e, g, h)$ be a public key. Given encryptions C_1 and $C_2 \in \mathbb{G}_1$ of messages m_1 and $m_2 \in \{0, 1, \dots, T\}$ respectively, anyone can create a uniformly distributed encryption of $m_1 + m_2 \bmod n$ by computing the product $C = C_1 C_2 h^r$ for a random number $r \in \{0, 1, \dots, n-1\}$. More importantly, anyone can multiply two encrypted messages once using the bilinear map. Set $g_1 = e(g, g)$ and $h_1 = e(g, h)$. Then g_1 is of order n and h_1 is of order q_1 . Also, write $h = g^{\alpha q_2}$ for some (unknown) $\alpha \in \mathbb{Z}$. Suppose we are given two ciphertexts $C_1 = g^{m_1} h^{r_1} \in \mathbb{G}$ and $C_2 = g^{m_2} h^{r_2} \in \mathbb{G}$. To build an encryption of product $m_1 \cdot m_2 \bmod n$ given only C_1 and C_2 , 1) select random $r \in \mathbb{Z}_n$, and 2) set $C = e(C_1, C_2) h_1^r \in \mathbb{G}_1$. Then

$$\begin{aligned} C &= e(C_1, C_2) h_1^r = e(g^{m_1} h^{r_1}, g^{m_2} h^{r_2}) h_1^r \\ &= g_1^{m_1 m_2} h_1^{m_1 r_2 + r_2 m_1 + q_2 r_1 r_2 \alpha + r} = g_1^{m_1 m_2} h_1^{r'} \in \mathbb{G}_1 \end{aligned}$$

where $r' = m_1 r_2 + r_2 m_1 + q_2 r_1 r_2 \alpha + r$ is distributed uniformly in $\mathbb{Z}_n$ as required. Thus, C is a uniformly distributed encryption of $m_1 m_2 \bmod n$, but in the group $\mathbb{G}_1$ rather than $\mathbb{G}$ (this is why we allow for just one multiplication). We note that the system is still additively homomorphic in $\mathbb{G}_1$. For simplicity, in this paper we denote an encryption of message m in $\mathbb{G}$ as $E_G(m)$ and one in $\mathbb{G}_1$ as $E_{G_1}(m)$.

2.3 Key sharing

In [2], efficient protocols are presented for a number of players to generate jointly RSA modulus $N = pq$ where p and q are prime, and each player retains a share of N . In this protocol, none of the players can know the factorization of N . They then show how the players can proceed to compute a public exponent e and the shares of the corresponding private exponent. At the end of the computation the players are convinced that N is a product of two large primes by using zero-knowledge proof. Their protocol was based on the threshold decryption that m out of m players can decrypt the secret. The cost of key generation for the shared RSA private key is approximately 11 times greater than that for simple RSA key generation. However the cost for computation is still practical. We use this protocol to share private keys among auction managers.

3 New efficient auction protocol

In this section, we show an efficient $M + 1st$ price auction based on bit-slice auction protocols.

3.1 Proposed $M + 1st$ price auction protocol

We define three types of player's status on i -th bit as W_j (Winner), C_j (Candidate) and S_j (Survivor) shown as below and the numbers of players in W_j and S_j as $|W_j|$ and $|S_j|$.

the Definition of Status:

$W_j[1...m]$: $W_j[i]=1$ if player P_i is decided to be a winner by upper j bits of the bids.

$C_j[1...m]$: $C_j[i]=1$ if player P_i is not decided to be a winner but has a possibility of $M + 1st$ highest bidder by upper j bits of the bids.

$S_j[1...m]$: $S_j[i]=1$ if $C_j[i]=1$ and j -th bit of P_i 's bid is 1.

Let $|W_j|$ and $|S_j|$ be the number of 1's in W_j and S_j , respectively.

Suppose that $B_{M+1st} = (b_{M+1st}^{(k-1)}, ..., b_{M+1st}^{(0)})_2$ is the $M + 1st$ highest bidding price and a bid of a player i is $Z_i = (z_i^{(k-1)}, ..., z_i^{(0)})_2$, where $()_2$ is the binary expression. The winner and winning price are found by the following protocol.

As initial setting, we set $W_k[i]=0$ ($1 \leq i \leq m$) and $C_k[i]=1$ ($1 \leq i \leq m$).

For $j = k-1$ to 0

$$S_j[i] = C_{j+1}[i] * z_i^{(j)} \quad (1 \leq i \leq m)$$

if $|W_{j+1}| + |S_j| > M$ then

$$b_{M+1st}^{(j)} = 1$$

$$C_j[i] = S_j[i] \quad (1 \leq i \leq m)$$

$$W_j[i] = W_{j+1}[i] \quad (1 \leq i \leq m)$$

else

$$b_{M+1st}^{(j)} = 0$$

$$W_j[i] = W_{j+1}[i] + S_j[i] \quad (1 \leq i \leq m)$$

$$C_j[i] = C_{j+1}[i] - S_j[i] \quad (1 \leq i \leq m)$$

end

If the number of Winners on $(j+1)$ -th bit and Survivors

on j -th bit is more than M , we keep Winners remained and update Candidates to eliminate players i in a set of $(C_j[i] - S_j[i])$, because they have no possibility to win the auction.

If the number of Winners on $(j+1)$ -th bit and Survivors on j -th bit is less than or equal to M , Survivors on j -th bid are determined as Winners, so we update W_j as $W_{j+1}[i] + S_j[i]$ and eliminate players i of $S_j[i]$ from $C_{j+1}[i]$.

3.2 Secure $M + 1st$ price auction using 2-DNF scheme and mix-and-match protocol

We assume n players, $P_1, ..., P_n$ and a set of auction managers, AM . The players bid their encrypted prices and publish them. The AM runs an auction with the encrypted bids and after the auction jointly decrypts the results of the protocol. Players find the winning price (the $M + 1st$ price) and the winners by decrypting the results. To maintain secrecy of the players' bidding prices through the protocol, we need to use the mix-and-match protocol. Here, we define three types of new tables, MAP_1 and MAP_2 . In the proposed protocol, the MAP_1 and MAP_2 tables are created among AM before an auction. The AM jointly computes values in the mix-and-match table for distributed decryption of plaintext equality test. The function of table MAP_1 , shown in Table 2, is a mapping $x_1 \in \{E_{G_1}(0), E_{G_1}(1)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$. The table MAP_2 , shown in Table 3, is the one for mapping $x_1 \in \{E_{G_1}(0), E_{G_1}(1), ..., E_{G_1}(m)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$. These tables can be constructed using the mix-and-match protocol because the Boneh-Goh-Nissim encryption has homomorphic properties.

3.2.1 Setting

AM jointly generates and shares private keys among themselves using the technique described in [2].

3.2.2 Bidding Phase

Suppose that $B_{M+1st} = (b_{M+1st}^{(k-1)}, ..., b_{M+1st}^{(0)})_2$ is the $M + 1st$ highest bidding price and a bid of a player i is $Z_i = (z_i^{(k-1)}, ..., z_i^{(0)})_2$, where $()_2$ is the binary expression. Each player P_i computes a ciphertext of his bidding price, Z_i , as

$$ENC_i = (b_i^{k-1}, ..., b_i^0)$$

where $b_i^j \in E_G(z_i^{(j)})$, and publishes ENC_i on the bulletin board. He also proves in zero-knowledge that $z_i^{(j)} = 0$ or 1 by using the technique described in [3].

3.2.3 Opening phase

Let $C_k = (c_1^k, ..., c_m^k)$, where each $c_i^k \in E_G(1)$ and $W_k = (w_1^k, ..., w_m^k)$, where each $w_i^k \in E_{G_1}(0)$.

(Step 1) For $j = k-1$ to 0, perform the following.

(Step 1-a) For $C_j = (c_1^j, ..., c_m^j)$, AM computes $s_i^j = Mul(b_i^j, c_i^j)$ for each player i , and

$$S_j = (Mul(c_1^j, b_1^j), ..., Mul(c_m^j, b_m^j))$$

Table 2: Table for MAP_1

x_1	x_2
$a_1 \in E_{G_1}(0)$	$b_1 \in E_G(0)$
$a_2 \in E_{G_1}(1)$	$b_2 \in E_G(1)$

Table 3: Table for MAP_2

x_1	x_2
$a_1 \in E_{G_1}(0)$	$b_1 \in E_G(0)$
$a_2 \in E_{G_1}(1)$	$b_2 \in E_G(0)$
$\dots$	$b_i \in E_G(0)$
$a_{M+1} \in E_{G_1}(M)$	$b_{M+1} \in E_G(0)$
$a_{M+2} \in E_{G_1}(M+1)$	$b_{M+2} \in E_G(1)$
$\dots$	$b_i \in E_G(1)$
$a_{m+1} \in E_{G_1}(m)$	$b_{m+1} \in E_G(1)$

$$h_j = \text{Mul}(b_1^j, c_1^j) \otimes \dots \otimes \text{Mul}(b_m^j, c_m^j)$$

$$d_j = w_1^j \otimes \dots \otimes w_m^j$$

(Step 1-b) The AM uses table MAP_1 for s_i^j for each i and finds the values of $\tilde{s}_i^j$. Let $\tilde{S}_j = (\tilde{s}_1^j, \dots, \tilde{s}_m^j)$.

(Step 1-c) AM uses table MAP_2 for $d_j \otimes h_j$ and decrypts the output value. If the output value is 0, the number of winners and survivors are less than $M+1$. Then, AM updates

$$W_j = W_{j+1} + S_j = (w_1^{j+1} \otimes s_1^j, \dots, w_m^{j+1} \otimes s_m^j)$$

$$C_{j-1} = C_j - \tilde{S}_j = (c_1^j \otimes (\tilde{s}_1^j)^{-1}, \dots, c_m^j \otimes (\tilde{s}_m^j)^{-1})$$

$$b_{M+1st}^{(i)} = 0$$

If the output value is 1, then

$$W_j = W_{j+1} = (w_1^{j+1}, \dots, w_m^{j+1})$$

$$C_{j-1} = \tilde{S}_j = (\tilde{s}_1^j, \dots, \tilde{s}_m^j)$$

$$b_{M+1st}^{(i)} = 1$$

(Step 2) For the final $W_0 = (w_1^0, \dots, w_m^0)$, AM decrypts each w_i^0 with verification protocols and obtains the winners of the auction. P_i is the winners if and only if plaintexts of $w_i^0 = 1$ and $\sum w_i^0 = M$. The $M+1st$ highest price is obtained as $B_{M+1st} = (b_{M+1st}^{(k-1)}, \dots, b_{M+1st}^{(0)})_2$.

Verification protocols

Verification protocols are the protocols for players to confirm that AM decrypts the ciphertext correctly. By using the protocols, each player can verify the results of the auction are correct. We denote b as a plaintext and C as a BGN encryption of b ($C = g^b h^r$), where g, h and r are elements used in BGN scheme and $f = C(g^b)^{-1}$. Before a player verifies whether b is the plaintext of C , the player must prove that a challenge ciphertext $C' = g^x f^r$ is created by himself with zero-knowledge proof that he has the value of x .

1. A player proves that he has random element $x \in \mathbb{Z}_n^*$ with zero-knowledge proof.
2. The player computes $f = C(g^b)^{-1}$ from the published values, h , g and b , and select a random

integer $r \in \mathbb{Z}_n^*$. He sends $C' = g^x f^r$ to AM .

3. The AM decrypts C' and sends value x' to the player.
4. The player verifies whether $x = x'$. AM can decrypt C' correctly only if $\text{order}(f) = q_1$, which means that the AM correctly decrypts C and publishes b as the plaintext of C .

3.3 Security

1. Privacy for bidding prices

Each player can not retrieve any information except for the winners and the $M+1st$ highest price. An auction scheme is secure if there is no polynomial time adversary that breaks privacy with non-negligible advantage $\epsilon(\tau)$. We prove that the privacy for bidding prices in the proposed auction protocols under the assumption that BGN encryption with the mix-and-match oracle is semantically secure. Given a message m , the mix-and-match oracle receives an encrypted value $x_1 \in E_{G_1}(m)$ and returns the encrypted value $x_2 \in E_G(m)$ according to the mix-and-match table shown in Table 3. (which has the same function as MAP_2). Given a message m and the ciphertext $x_1 \in E_{G_1}(m)$, the function of mix-and-match table is to map $x_1 \in E_{G_1}(m) \rightarrow x_2 \in E_G(m)$. The range of the input value is supposed to be $\{0, 1, \dots, m\}$ and the range of the output is $\{0, 1\}$. We do not consider cases where the input values are out of the range. Using this mix-and-match oracle, an adversary can compute any logical function without the limit where BGN encryption scheme can use only one multiplication on encrypted values. MAP_1 can also be computed if the range of the input value is restricted in $\{0, 1\}$. Here, we define two semantic secure games and advantages for BGN encryption scheme and the proposed auction protocols. We also show that if there is adversary $\mathcal{B}$ that breaks the proposed auction protocol, we can compose adversary $\mathcal{A}$ that breaks the semantic security of the BGN encryption with the mix-and-match oracle by using $\mathcal{B}$.

Definition 1

Let $\Pi = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ be a BGN encryption scheme, and let $A^{O_1} = (A_1^{O_1}, A_2^{O_1})$, be a probabilistic polynomial-time algorithm, that can use the mix-and-match oracle O_1 .

$$\text{BGN-Adv}(\tau) = \Pr[\text{EXPT}_{A, \Pi}(\tau) \Rightarrow 1] - 1/2$$

where, $\text{EXPT}_{A, \Pi}$ is a semantic security game of the BGN encryption scheme with the mix-and-match oracle shown in Fig. 1.

$(PK, SK) \leftarrow \text{KeyGen}$ $(m_0, m_1, s) \leftarrow A_1^{o_1}(PK)$ $b \leftarrow \{0, 1\}$ $c \leftarrow \text{Encrypt}(PK, m_b)$ $b' \leftarrow A_2^{o_1}(c, s)$ $\text{return } 1 \text{ iff } b = b'$

Figure 1: $EXPT_{A,\Pi}$

We then define an adversary $\mathcal{B}$ for an auction protocol and an advantage for $\mathcal{B}$.

Definition 2

Let $\Pi = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ be a BGN encryption scheme, and let B be two probabilistic polynomial-time algorithm B_1 and B_2 .

$$\text{Auction-Adv}(\tau) = \Pr[EXPT_{B,\Pi} = 1] - 1/2$$

where $EXPT_{B,\Pi}$ is a semantic security game of the privacy of the auction protocol shown in Fig. 2.

First of all, B_1 generates k -bit integers, $b_1, b_2, \dots, b_{m-1}$ as plaintexts of bidding prices for player 1 to $m-1$, and two challenge k -bit integers as b_{m_0}, b_{m_1} where b_{m_0} and b_{m_1} are the same bits except for i -th bit m_0^i and m_1^i . We assume b_{m_0} and b_{m_1} are not the $M+1$ st highest price. Then the auction is executed with $(\text{Encrypt}(PK, b_1), \text{Encrypt}(PK, b_2), \dots, \text{Encrypt}(PK, b_{m-1}), \text{Encrypt}(PK, b_{m_b}))$ as the players' encrypted bidding prices where $b \xleftarrow{r} \{0,1\}$. After the auction, B_2 outputs $b' \in \{0,1\}$ as a guess for b . $\mathcal{B}$ wins if $b = b'$.

Theorem 1 *The privacy of the auction protocols is secure under the assumption that the BGN encryption is semantically secure with a mix-and-match oracle.*

We show if there is adversary $\mathcal{B}$ that breaks the security of the proposed auction protocol, we can compose adversary $\mathcal{A}$ that breaks the semantic security of the BGN encryption with the mix-and-match oracle. $\mathcal{A}$ receives two challenge k -bit integers as b_{m_0} and b_{m_1} from $\mathcal{B}$ and then $\mathcal{A}$ uses m_0^i and m_1^i as challenge bits for the challenger of the BGN encryption. Then $\mathcal{A}$ receives $\text{Encrypt}(PK, m_b^i)$ and executes a secure auction protocol with the mix-and-match table. In the auction, when decrypted values are needed, $\mathcal{A}$ can calculate them since he knows all the input values, $b_1, b_2, \dots, b_{m-1}$ except the i -th bit of b_{m_b} . Through the protocol, $\mathcal{B}$ observes the calculation of the encrypted bids and the results of the auction. After the auction, $\mathcal{B}$ outputs b' , which is the guess for b . $\mathcal{A}$ outputs b' , which is the same guess with $\mathcal{B}$'s output for b_{m_b} . If $\mathcal{B}$ can break the

privacy of the bidding prices in the proposed auction protocol with advantage $\epsilon(\tau)$, $\mathcal{A}$ can break the semantic security of the BGN encryption with the same advantage.

2. Correctness

For correct players' inputs, the protocol outputs the correct winner and price. From Theorem 1 introduced in Section 1.4, the bit-slice auction protocol obviously satisfies the correctness.

3. Verification of the evaluation

To verify whether the protocol works, players need to validate whether the AM decrypts the evaluations of the circuit on ciphertexts through the protocol. We use the verification protocols introduced above so that each player can verify whether the protocol is computed correctly.

4 Conclusion

We introduced new efficient secure $M+1$ st price auction protocols based on the mix-and-match protocol and the BGN encryption. As a topic of future work, we will try to compose a secure auction protocol without using the mix-and-match protocol.

References

- [1] M. Abe and K. Suzuki, "M + 1st price auction using homomorphic encryption", Proceedings of Public Key Cryptography 2002, LNCS Vol.2274, pp 115-124.
- [2] D. Boneh and M. Franklin, "Efficient Generation of Shared RSA keys", Invited paper Public Key Cryptography 1998, LNCS Vol.1431, pp. 1-13.
- [3] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts", Proceedings of Theory of Cryptography (TCC) 2005, LNCS Vol.3378, pp. 325-341.
- [4] D. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, ACM 1981, pp 84-88.
- [5] M. K. Franklin and M. K. Reiter, "The design and implementation of a secure auction service", IEEE Transactions on Software Engineering, Vol.22, No.5, 1995, pp.302-312.
- [6] M.Jakobsson and A.Juels, "Mix and Match: Secure Function Evaluation via Ciphertexts", Proceedings of Asiacrypt 2000, LNCS Vol. 1976, pp. 162-177.
- [7] A. Juels and M. Szydlo, "A Two-Server Sealed-Bid Auction Protocol", Proceedings of Financial Cryptography 2002, LNCS Vol. 2357, pp. 72-86.
- [8] K. Kurosawa and W. Ogata, "Bit-Slice Auction Circuit", Proceedings of the 7th European Symposium on Research in Computer Security 2002, LNCS Vol.2502, pp. 24-38.

$ \begin{aligned} (PK, SK) &\leftarrow KeyGen \\ (b_1, b_2, \dots, b_{m-1}, b_{m0}, b_{m1}, s) &\leftarrow B_1(PK) \\ b &\leftarrow \{0, 1\} \\ c &\leftarrow (Encrypt(PK, b_1), Encrypt(PK, b_2), \dots, Encrypt(PK, b_{m-1}), Encrypt(PK, b_{mb})) \\ &\text{execute auction protocols using} \\ &\text{c as players' bids} \\ b' &\leftarrow B_2(c, s) \\ &\text{return 1 iff } b = b' \end{aligned} $

Figure 2: $EXPT_{B,\Pi}$

- [9] H. Lipmaa, N. Asokan, and V. Niemi. "Secure Vickrey auctions without threshold trust", Proceedings of the 6th Annual Conference on Financial Cryptography, LNCS Vol.2357, pp. 87-101.
- [10] M. Naor, B. Pinkas, and R. Sumner. "Privacy preserving auctions and mechanism design" Proceedings of the 1st ACM Conference on Electronic Commerce (ACM-EC), ACM press 1999, pp.129-139.
- [11] T. Mitsunaga, Y. Manabe, and T. Okamoto. "Efficient Secure Auction Protocols Based on the Boneh-Goh-Nissim Encryption" Proceedings of 5th International Workshop on Security 2010, LNCS Vol. 6434, pp. 149-163.
- [12] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", Proceedings of Eurocrypt 1998, LNCS Vol. 1403, pp. 308-318.
- [13] P.Pallier, "Public-key cryptosystems based on composite degree residuosity classes", Proceedings of Eurocrypt 1999, LNCS Vol. 1592, pp. 223-238.
- [14] C. Park, K. Itoh, and K.Kurosawa "All/nothing election scheme and anonymous channel", Proceedings of Eurocrypt 1993, LNCS Vol. 765, pp. 248-259.
- [15] Y. Tamura, T. Shiotsuki, and A. Miyaji, "Efficient Proxy-bidding system", IEICE Transactions on Fundamentals. Vol. J87-A, No.6(2004), 835-842.