SCIS 2010 The 2010 Symposium on Cryptography and Information Security
Takamatsu, Japan, Jan. 19-22, 2010
The Institute of Electronics, Information and Communication Engineers

Efficient Secure Auction Protocols Based on the Boneh-Goh-Nissim Encryption

Takuho Mitsunaga * Yoshifumi Manabe † Tatsuaki Okamoto †

Abstract— This paper presents efficient secure auction protocols for first price auction and second price auction. Previous auction protocols are based on a generally secure multi-party protocol called mix-and-match protocol. However, the time complexity of the mix-and-match protocol is large, though it can securely calculate any logical circuits. Our auction protocols reduce the number of times the mix-and-match protocol is used by replacing them with the Boneh-Goh-Nissim encryption that enables calculation of 2-DNF of encrypted data.

Keywords: secure auction, Boneh-Goh-Nissim encryption, mix-and-match protocol, 2-DNF

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). Jakobsson and Juels proposed a secure MPC protocol to evaluate a function comprising a logical circuit, called mix-and-match [6]. Based on the mix-and-match protocol, we can easily find a secure auction protocol by repeating the millionaires' problem for two players. However, the mix-and-match protocol requires two plaintext equity tests for a two-input one-output gate. Thus, it is important to reduce the number of gates in C_f to achieve function f. Kurosawa and Ogata suggested a more efficient auction protocol than that one based on the millionaire's problem [7].
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 [10], so it has additive homomorphism and in addition the bilinear map allows one multiplication on encrypted values. As a result, its property allows the evaluation of multivaliate polynomials of a total of degree two on encrypted values.

1.2 Our result

In this paper, we introduce bit-slice auction protocols based on the public evaluation of the 2-DNF formula. For the first price auction, the protocol uses no mix-and-match gates. For the second price auction, we use the mix-and-match protocol fewer times than that suggested in [7].

1.3 Related works

There are many auction protocols[1, 5, 8], however, they have problems such as those described hereafter. The first secure auction scheme proposed by Franklin and Reiter [5] does not provide full privacy, since at the end of an auction players can know other player bids. Lipmaa, Asokan and Niemi proposed an efficient secure auction scheme [8]. In this scheme, the trusted auction authority can know the bid statistics. Abe and Suzuki suggested a secure auction scheme for the $M+1st$ auction based on homomorphic encryption [1]. The $M+1st$ price auction is a generalized auction protocol for selling M units of a single kind of goods, and the $M+1st$ highest price is the winning price, however a player's bid is not a binary expression. So, it requires a cost of $O(m2^k)$ for a m-player and k-bid auction.

2 Preliminaries

2.1 Mix and Match Protocol

The mix-and-match protocol is a general multiparty protocol proposed by [6]. It uses a homomorphic encryption scheme and a MIX net. This model involves n players, denoted by $P_1, P_2, ..., P_n$ 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 $f(B_1, ..., B_n)$ 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. She proves that each 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 $f(B_1, ..., B_n)$. They jointly decrypt this ciphertext value to reveal the output of the function f.

* Department of Social Informatics, Graduate School of Informatics, Kyoto Univ.
† NTT Labs, Nippon Telegraph and Telephone Corporation.

2.1.1 Requirements for the Encryption Function

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

Homomorphic property There exists a 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
$a \cdot e = \underbrace{c \otimes c \otimes \cdots \otimes c}_{a}$

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 $E(m)$.

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

A MIX protocol (proposed in [4]) takes a list of ciphertexts, $(\xi_1, ..., \xi_L)$ and outputs a permuted and re-encrypted list of the ciphertexts $(\xi'_1, ..., \xi'_L)$ without revealing the relationship between $(\xi_1, ..., \xi_L)$ and $(\xi'_1, ..., \xi'_L)$, where ξ_i or ξ'_i can be a single ciphertext c, or a list of l ciphertexts, $(c_1, ..., c_l)$, for some $l > 1$. For all players to verity the validity of $(\xi'_1, ..., \xi'_L)$, we use the universal verifiable MIX net protocol suggested by [11].

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 $c_0 = c_1 \otimes c_2^{-1}$.
(Step 1) For each player P_i (where i = 1,...,m):
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.
(Step 2) Let $z = z_1 \otimes z_2 \otimes \cdots \otimes z_n$. The players jointly decrypt z using threshold verifiable decryption and obtain plaintext v. Then it holds that

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

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 of 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 ith row be (a'_i, b'_i, c'_i) for i = 1,...,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.

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

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

2.2 Bit-Slice Auction Circuit

We introduce an efficient auction circuit called the bit-slice auction circuit suggested by [6].
Suppose that $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$ is the highest bidding price and a bid of a player i is $B_i = (b_i^{(k-1)}, ..., b_i^{(0)})_2$, where $()_2$ is the binary expression. Then the proposed circuit first determines $b_{max}^{(k-1)}$ by evaluating the most significant bits of all the bids. It next determines $b_{max}^{(k-2)}$ by looking at the second most significant bits of all the bids, and so on.
For two m-dimensional binary vectors
$\mathbf{X} = (x_1, ..., x_m)$ and $\mathbf{Y} = (y_1, ..., y_m)$,

$$\mathbf{X} \wedge \mathbf{Y} = (x_1 \wedge y_1, ..., x_m \wedge y_m)$$

Let D_j be the highest price when considering the upper j bits of the bids. That is,

$$\begin{gathered} D_1 = (b_{max}^{(k-1)}, 0, ..., 0)_2 \\ D_2 = (b_{max}^{(k-1)}, b_{max}^{(k-2)}, 0, ..., 0)_2 \\ \cdots \\ D_k = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2 \end{gathered}$$

In the j-th round, we find $b_{max}^{(k-1)}$ and eliminate a player P_i such that his bid satisfies $B_i < D_1$. For example, in the case of $j = 1$, a player i is eliminated if his bid $B_i < D_j$. By repeating this operation for 1 to $k - 1$, at the end the remaining bidder is the winner.
For this purpose, we update $\mathbf{W} = (w_1, ..., w_m)$ such that

$$w_i = \begin{cases} 1 & if\ B_i \geq D_j \\ 0 & otherwise \end{cases}$$

for $j = 1$ to k. The circuit is obtained by implementing the following algorithm. For given m bids, $B_1, ..., B_m$, V_j is defined as

$$V_i = (b_1^{(j)}, ..., b_m^{(j)})$$

for j = 0,...,$k - 1$, that is, V_j is the vector consisting of the (j+1)th lowest bit of each bid. Let $\mathbf{W} = (w_1, ..., w_m)$, where each $w_j = 1$. For $j = k - 1$ to 0, perform the following;
(Step 1) For $\mathbf{W} = (w_1, ..., w_m)$, let

$$\begin{gathered} S_j = \mathbf{W} \wedge V_j \\ = (w_1 \wedge b_1^{(j)}, ..., w_m \wedge b_m^{(j)}) \\ b_{max}^{(j)} = (w_1 \wedge b_1^{(j)}) \vee \cdots \vee (w_m \wedge b_m^{(j)})\ . \end{gathered}$$

(Step 2) If $b_{max}^{(j)} = 1$, then let $\mathbf{W} = S_j$.

Then the highest price is obtained
as $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$. Let the final $\mathbf{W}$ be $(w_1, ..., w_m)$. Then P_i is the winner if and only if $w_i = 1$. We summarize the algorithm as the following theorem.

Theorem 1 *[7] In the bit-slice auction above,*
- B_{max} is the highest bidding price.
- For the final $\mathbf{W} = (w_1, ..., w_m)$, P_i is a winner if and only if $w_i = 1$ and P_i is the only player who bids the highest price B_{max}.

2.3 Evaluating 2-DNF Formulas on Ciphertexts

Given encrypted Boolean variables $x_1, ..., x_n \in \{0, 1\}$, 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.

2.3.1 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$.

2.3.2 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is a bilinear map. On input τ, algorithm $\mathcal{G}$ works as indicated below,

1. Generate two random bit primes, q_1, 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ 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.

2.3.3 Homomorphic public key system

We now describe the proposed public key system which resembles the Paillier [10] and the Okamoto-Uchiyama encryption schemes [9]. We describe the three algorithms comprising the system.

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 $SK = q_1$.

Encrypt(PK, M) We assume that the message space consists of integers in set
$\{0, 1, ..., T\}$ with $T < q_2$. We encrypt binary representation of bids in our main application, in which case $T = 1$. To encrypt a message m using public key PK, select a random number $r \in \{0, 1, ..., n-1\}$ and compute $C = g^m h^r \in \mathbb{G}$.
Output C as the ciphertext.

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 $\hat{g}$.

2.3.4 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 $m_2 \in \{0, 1, ..., T\}$ 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 $r \in \{0, 1, ..., n-1\}$. 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) random $r \in \mathbb{Z}_n$, and 2) set $C = e(C_1, C_2) h_1^r \in \mathbb{G}_1$. Then

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

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.

2.4 Key sharing

In [2], efficient protocols are presented for a number of players to generate jointly an RSA modulus $N = pq$ where p and q are prime, and each player retain a share of N. In this protocol, none of the players can know the factorization of N.Their protocol was based on the threshold decryption that m out of m players can decrypt the secret. We use this protocol to share private keys among auction managers.

3 New efficient auction protocol

In this section, we show bit-slice auction protocol based on the evaluation of multivariate polynomials of a total degree two on encrypted values.

3.1 First price auction with 2-DNF scheme

We assume n players, $P_1, ..., P_n$ and a set of auction managers, AM. The players bid their encrypted prices, and through the protocol they publish encrypted flags whether they are still in the auction. AM jointly decrypts the result of the protocol. Players find the highest price through the protocol and the winner by decrypting the result.

3.1.1 Setting

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

3.1.2 Bidding Phase

Each player P_i computes a ciphertext of his bidding price, B_i as

$$ENC_i = (c_{i,k-1}, ..., c_{i,0})$$

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

3.1.3 Opening Phase

Suppose that $c_1 = g^{b_1}h^{r_1} \in E_G(b_1)$ and $c_2 = g^{b_2}h^{r_2} \in E_G(b_2)$, where b_1, b_2 are binary and $r_1, r_2 \in \mathbb{Z}_p^*$ are random numbers. We define two polynomial time computable operations Mul and $\otimes$ by applying a 2DNF formula for AND, OR respectively.

$$Mul(c_1, c_2) = e(c_1, c_2) = e(g^{b_1}h^{r_1}, g^{b2}h^{r_2}) \in E_{G_1}(b_1 \wedge b_2)$$
$$c_1 \otimes c_2 = g^{b_1}h^{r_1} \cdot g^{b2}h^{r_2} = g^{b_1+b_2}h^{r_1+r_2} \in E_G(b_1 + b_2)$$

by applying a 2DNF formula for AND.
AM generates $W = (w_1, ..., w_m)$, where each w_j =1, and encrypts them as $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$. AM shows that $\widetilde{W}$ is the encryption of (1,...,1) with the verification protocols.
(Step 1) For $j = k$ -1 to 0, perform the following.
(Step 1-a) For $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM computes $s_{i,j} = Mul(\tilde{w}_i, c_{i,j})$ for each player i, and

$$S_j = (Mul(\tilde{w}_1, c_{1,j}), ..., Mul(\tilde{w}_m, c_{m,j}))$$
$$h_j = Mul(\tilde{w}_1, c_{1,j}) \otimes \cdots \otimes Mul(\tilde{w}_m, c_{m,j})$$

(Step 1-b) AM takes a plaintext equality test regarding whether h_j is an encryption of 0. If h_j is an encryption of 0, AM publishes 0 as the value of $b_{max}^{(j)}$ and proves it with the verification protocols, otherwise, AM publishes 1 as the value of $b_{max}^{(j)}$.
(Step 1-c) If $b_{max}^{(j)} = 1$, then each player creates a new encryption $\tilde{w}_i$ which has the same plaintext value of $s_{i,j}$, otherwise he uses w_i for the next bit. And the player shows the validity of computation with zero-knowledge proof.
(Step 2) For the final $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM decrypts each $\tilde{w}_i$ with the verification protocols and obtains plaintext w_i. The highest price is obtained as
$B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$. P_i is a winner if and only if $w_i = 1$.

3.2 Second price auction with 2-DNF scheme and mix-and-match protocol

In the second price auction, the information that players can find is the second highest price and the bidder of the highest price. To maintain secrecy of the highest bid through the protocol, we need to use the mix-and-match protocol. However, we can reduce the number of times we use it. As a result, the proposed protocol is more efficient than that in [7]. Here, we define three types of new tables, $Select_m$, MAP_1 and MAP_2 for the second price auction. In our protocol, mix-and-match tables are created among AM before an auction. AM computes jointly for distributed decryption of plaintext equality test. Table $Select_m$ is also used for the second price auction protocol in [7]; MAP_1 and MAP_2 are new tables that we propose. MAP_1 and MAP_2 are used for mapping a encrypted value $a_1 \in E_{G_1}$ (which is an output of a computation with one multiplication) to $a_2 \in E_G$
Table $Select_m$ has $2k + 1$ input bits and k output bits as follows.

$$Select_m(b, x^{(m-1)}, ..., x^{(0)}, y^{(m-1)}, ..., y^{(0)})$$
$$= \begin{cases} (x^{(m-1)}, ..., x^{(0)}) & if\ b = 1 \\ (y^{(m-1)}, ..., y^{(0)}) & otherwise \end{cases}$$

For two encrypted input vectors $(x^{(k-1)}, ..., x^{(0)})$ and $(y^{(k-1)}, ..., y^{(0)})$, b is an encrytption of check bit that selects which vector to output,
$(x^{(k-1)}, ..., x^{(0)})$ or $(y^{(k-1)}, ..., y^{(0)})$. For secure computation, AM re-encrypts an output vector.
The function of table MAP_1 is a mapping
$x_1 \in \{E_{G_1}(0), E_{G_1}(1)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$.

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

The table MAP_2 is the one for mapping
$x_1 \in \{E_{G_1}(0), E_{G_1}(1), ..., E_{G_1}(m)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$.

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

These tables can be composed on using the mix-and-match protocol because the Boneh-Goh-Nissim encryption has homomorphic properties. The setting and bidding phases are the same as the first price auction, so we start from the opening phase.

3.2.1 Opening Phase

Let $\widetilde{W} = (\tilde{w}_1, .., \tilde{w}_m)$, where each $w_j \in E_G(1)$ shown above.
(Step 1) For $j = k$ -1 to 0, perform the following.
(Step 1-a) For $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM computes $s_{i,j} = Mul(\tilde{w}_i, e_{i,j})$ for each player i, and

$$S_j = (Mul(\tilde{w}_1, e_{1,j}), ..., Mul(\tilde{w}_m, e_{m,j}))$$
$$h_j = Mul(\tilde{w}_1, e_{1,j}) \otimes \cdots \otimes Mul(\tilde{w}_m, e_{m,j})$$

(Step 1-b) AM uses table MAP_1 for $s_{i,j}$ for each i and find the values of $\tilde{s}_{i,j}$. Let $\widetilde{S}_j = (\tilde{s}_{1,j}, ..., \tilde{s}_{m,j})$. AM also uses the table MAP_2 for h_j as an input value. By using this table, AM retrieve $E(b_j) \in E_G(0)$ if h_j is a ciphertext of 1, otherwise he retrieves $E_{b_j} \in E_G(1)$.
(Step 1-c) AM uses the table $Select_m$ as input values $(E(b_j), \widetilde{S}_j, W)$.
By using table $Select_m$, if $E(b_j)$ is the encryption of 1, AM updates $W = \widetilde{S}_j$, otherwise W remains unchanged.
(Step 2) For the final $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM decrypts each $\tilde{w}_i$ with verification protocols and obtains the plaintext w_i. P_i is a winner if and only if $w_i = 1$. We remove the player who bids the highest price and run the first price auction protocol again. The second highest price is obtained as $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$.
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 result of auction is correct. Denote b is a palintext and C is a BGN encryption of b ($C = g^b h^r$), where g, h, r is elements used in BGN scheme and $f = (h)(g^b)^{-1}$. Before a player verifies

whether b is the plaintext of C, the player has to 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 $x \in Z_n^*$ with zero-knowledge proof.
2. The player computes $f = (h)(g^b)^{-1}$ from published the 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. AM decrypts C and sends value x' to the player.
4. The player verifies whether $x = x'$. AM can decrypt C correctly only if order$(f) = q$, which means 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 the winner and the highest price or the second highest price(the first price auction, second price auction respectively). 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. The mix-and-match oracle receives an encrypted value $x_1 \in E_{G_1}$ and returns the encrypted value $x_2 \in E_G$ according to the mix-and-match table below (which has the same function with MAP_2).

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

   Given an encrypted value $x_1 \in E_{G_1}$, the function of mix-and-match table is a mapping $x_1 \in E_{G_1} \rightarrow x_2 \in E_G$. The range of input value is supposed to be from 0 to m and the one of output is from 0 to 1. We do not consider a case when an input value is out of the range. Using this mix-and-match oracle, an adversary can compute any logical function without the limit that BGN encyrption scheme can use only one multiplication on encrypted values. So, an adversary can calculate $Select_m(b, x^{(m-1)}, ..., x^{(0)}, y^{(m-1)}, ..., y^{(0)}) = b(x^{(m-1)}, ..., x^{(0)}) + (1-b)(y^{(m-1)}, ..., y^{(0)})$ with additional polynomial computation. MAP_1 is also can be composed if the range of input value is restricted 0 to 1. Here, we define two semantic secure games and advantages for BGN encryption scheme and our auction protocols and show if there is adversary $\mathcal{B}$ that breaks the proposed auction protocol, we can compose adversary $\mathcal{A}$ by $\mathcal{B}$.

   Definition 1
   Let Π = $(KeyGen, Encrypt(PK, M), Decrypt(SK, C))$ be a BGN encryption scheme, and let A^{O_1} be two probabilistic polynomial-time algorithms $A_1^{O_1}$ and $A_2^{O_1}$, that can use the mix-and-match oracle O_1.

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

   where, $EXPT_{A,\Pi}$ is a semantic security game of the BGN encryption scheme with mix-and-match oracle shown in Table 1. We then define an adversary $\mathcal{B}$ for an auction protocol and an advantage for $\mathcal{B}$.

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

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

   Definition 2
   Let Π = $(KeyGen, Encrypt, Decrypt)$ 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 Table 2. First of all, B_1 generates k-bit integers, $b_1, b_2, ..., b_{m-1}$ 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 and m_1. We assume b_{m_0} and b_{m_1} are not the first price bid in a first price auction and the second highest price in a second price auction.Then the auction is executed with
   $(Encrypt(PK, b_1), Encrypt(PK, b_2), ...,$
   $Encrypt(PK, b_{m-1}), Encrypt(PK, b_{mb}))$
   as players' bids as encrypted the players' 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 2 *The privacy of 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} and then $\mathcal{A}$ uses them as challenge bits for the challenger of the BGN encryption. Then $\mathcal{A}$ receives $Encrypt(PK, m_b)$ and executes a secure auction protocol with the mix-and-match table. In the auction, when decrytpted values are needed, $\mathcal{A}$ can calculates them since he knows the all input values, $b_1, b_2, ..., b_{m-1}$ except $i-th$ bit of b_{mb}. Through the protocol, $\mathcal{B}$ surveies the calculation of encrypted bids and the result 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 breaks the privacy of bidding prices in our auction protocol with advantage $\epsilon(\tau)$, $\mathcal{A}$ can break semantic security of 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 2.2, the bit-slice auction protocol obviously satisfies the correctness.

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

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

	AND	OR	$Select_m$	MAP_1	MAP_2	Total PET(approx.)
[KO02]	$(2m-1)k$	$(m-1)k$	k	0	0	$(13mk/2) - 4k$
Proposed	0	0	k	mk	k	$2mk$

Table 1: The number of PET in the second price auction

	AND	PET	Total PET(approx.)
[KO02]	mk	k	$2mk + k$
Proposed	0	k	k

Table 2: The number of PET in the first price auction

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

4 Comparison

4.1 First price auction

The protocol proposed [7] requires mk AND computations and k plaintext equality tests. One AND computation requires two plaintext equality tests. So, the total number of plaintext equality test is $mk+k$. On the other hand, the proposed protocol is based on only a 2-DNF scheme and requires k plaintext equality tests. A comparison between the proposed protocol and that in [7] is shown in the Table 2 above.

4.2 Second price auction

In the second price auction protocol, the protocol in [7] requires $(2m-1)$ AND, $(m-1)k$ OR and k $Select_m$ gates. One OR gate requires two plaintext equality tests and one $Select_m$ gate requires one test, so in total approximately $(13mk) - 4k$ requires plaintext equality tests are required. Conversely, the proposed protocol requires MAP_1 mk times and MAP_2 k times. MAP_1 requires one plaintext equality test and MAP_2 requires approximately one half of m times on average, so in total $2mk$. A comparison between the proposed protocol and that in [7] is shown in the Table 1 above.

5 Conclusion

We introduced new efficient auction protocols based on the BGN encryption and showed that they are approximately two fold more efficient than the proposed in [7]. 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. 129-140.

[7] 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.

[8] 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.

[9] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", Proceedings of Eurocrypt 1998, LNCS Vol. 1403, pp. 308-318.

[10] P.Pallier, "Public-key cryptosystems based on composite degree residuosity classes", Proceedings of Eurocrypt 1999, LNCS Vol. 1592, pp. 223-238.

[11] C. Park, K. Itoh, and K.Kurosawa "All/nothing election scheme and anonymous channel", Proceedings of Eurocrypt 1993, LNCS Vol. 765, pp. 248-259.