PAPER *Special Section on Cryptography and Information Security*

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

Takuho MITSUNAGA[†a)], ***Nonmember***, **Yoshifumi MANABE**[††b)], ***and*** **Tatsuaki OKAMOTO**[†††c)], ***Members***

SUMMARY 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 based on plaintext equality tests. However, the time complexity of the plaintext equality tests is large, although the mix-and-match protocol can securely calculate any logical circuits. The proposed protocols reduce the number of times the plaintext equality tests is used by replacing them with the Boneh-Goh-Nissim encryption, which enables calculation of 2-DNF of encrypted data.
key words: *secure auction, 1st price auction, 2nd 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). 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. However, the mix-and-match protocol requires two plaintext equality tests for a two-input one-output gate. Furthermore, one plaintext equality test requires one distributed decryption among players. Thus, it is important to reduce the number of gates in C_f to achieve function f.

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 [9].

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 Pallier'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 first and second price auction protocols based on BGN scheme introduced in [3] and show that we can reduce the number of plaintext equality tests in both protocols.

1.2 Related Works

As related works, there are many auction protocols, 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 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 [11]. 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.

Lipmaa, Asokan and Niemi proposed an efficient $M + 1st$ secure auction scheme [10]. The $M + 1st$ price auction is a type of sealed-bid auction for selling M units of a single kind of goods, and the $M + 1st$ highest price is the winning price. M bidders who bid higher prices than the winning price are winning bidders, and each winning bidder buys one unit of the goods at the $M + 1st$ winning price. 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 en-

Manuscript received March 23, 2012.
Manuscript revised July 3, 2012.
[†]The author is with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
[††]The author is with NTT Communication Science Laboratories, NTT Corporation, Musashino-shi, 180-8585 Japan.
[†††]The author is with NTT Secure Platform Laboratories, NTT Corporation, Musashino-shi, 180-8585 Japan.
a) E-mail: mitsunaga@ai.soc.i.kyoto-u.ac.jp
b) E-mail: manabe.yoshifumi@lab.ntt.co.jp
c) E-mail: okamoto.tatsuaki@lab.ntt.co.jp
DOI: 10.1587/transfun.E96.A.68

cryption [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 [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

In this paper, we introduce bit-slice auction protocols based on the public evaluation of the 2-DNF formula. In the both of first and second price auction, our protocols are more efficient than original protocol suggested in [9].

2. Preliminaries

2.1 The Model of Auctions and Outline of Auction Protocols

This model involves m players, denoted by $P_1, P_2, \ldots, P_m$ 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 $f(B_1, \ldots, B_m)$ without revealing any additional information. Its outline is as follows.

1. **Input stage:** Each $P_i (1 \leq i \leq m)$ 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_M, the players obtain O_M, a ciphertext encrypting $f(B_1, \ldots, B_M)$. 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 v by $E(v)$ and a particular encryption of v by $c \in E(v)$.

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 $c \in E(v)$, then $c^{-1} \in E(-v \bmod q)$.
2. If $c_1 \in E(v_1)$ and $c_2 \in E(v_2)$, then $c_1 \otimes c_2 \in E(v_1 + v_2 \bmod q)$.

For a positive integer a, define
$a \cdot c = \underbrace{c \otimes c \otimes \cdots \otimes c}_{a}$.

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

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

2.1.2 MIX Protocol

The MIX protocol [4] takes a list of ciphertexts, $(\xi_1, \ldots, \xi_L)$, and outputs a permuted and re-encrypted list of the ciphertexts $(\xi'_1, \ldots, \xi'_L)$ without revealing the relationship between $(\xi_1, \ldots, \xi_L)$ and $(\xi'_1, \ldots, \xi'_L)$, where ξ_i or ξ'_i can be a single ciphertext c, or a list of l ciphertexts, $(c_1, \ldots, c_l)$, for some $l > 1$. For all players to verity the validity of $(\xi'_1, \ldots, \xi'_L)$, 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 $i = 1, \ldots, m$): 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.
2. (Step 2) Let $z = z_1 \otimes z_2 \otimes \cdots \otimes z_m$. 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, m 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. m players first consider the standard encryption of each entry in the table shown in Table 1.
2. By applying a MIX protocol to the four rows of the table, m players jointly compute blinded and permuted rows of the table. Let the ith row be (a'_i, b'_i, c'_i) for $i = 1, \ldots, 4$.
3. m 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 $c'_i \in E(x_1 \wedge x_2)$.

Table 1 Mix-and-match table for AND.

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

2.2 Bit-Slice Auction Circuit

We introduce an efficient auction circuit called the bit-slice auction circuit described in [6]. In this scheme, we assume only one player bids the highest bidding price, so we do not consider a case more than one player becomes the winners. Suppose that $B_{max} = (b_{max}^{(k-1)}, \ldots, b_{max}^{(0)})_2$ is the highest bidding price and a bid of a player i is $B_i = (b_i^{(k-1)}, \ldots, b_i^{(0)})_2$, 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 $\mathbf{X} = (x_1, \ldots, x_m)$ and $\mathbf{Y} = (y_1, \ldots, y_m)$,

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

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

$$D_1 = (b_{max}^{(k-1)}, 0, \ldots, 0)_2$$
$$D_2 = (b_{max}^{(k-1)}, b_{max}^{(k-2)}, 0, \ldots, 0)_2$$
$$\ldots$$
$$D_k = (b_{max}^{(k-1)}, \ldots, b_{max}^{(0)})_2$$

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

For this purpose, we update $\mathbf{W} = (w_1, \ldots, w_m)$ 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, $B_1, \ldots, B_m$, V_j is defined as

$$V_j = (b_1^{(j)}, \ldots, b_m^{(j)})$$

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

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

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

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

Property 1: [9] In the bit-slice auction above,
- B_{max} is the highest bidding price.
- For the final $\mathbf{W} = (w_1, \ldots, w_m)$, 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 $x_1, \ldots, x_n \in \{0, 1\}$, 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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$.

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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ 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.

2.3.3 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 $u \stackrel{R}{\leftarrow} \mathbb{G}$ 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 $\{0, ..., T\}$ with $T < q_2$. We encrypt the binary representation of bids in our main application, in the case $T = 1$. To encrypt a message $M = v$ using public key PK, select a random number $r \in \{1, ..., n-1\}$ and compute

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

Output C as the ciphertext.

3.Decrypt(SK, C) To decrypt a ciphertext C using the private key $SK = q_1$, observe that $C^{q_1} = (g^v h^r)^{q_1} = (g^{q_1})^v$. 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 v_1 and $v_2 \in \{0, 1, ..., T\}$ respectively, anyone can create a uniformly distributed encryption of $v_1 + v_2 \bmod n$ by computing the product $C = C_1 C_2 h^r$ for a random number $r \in \{1, ..., n-1\}$. 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 $C_1 = g^{v_1} h^{r_1} \in \mathbb{G}$ and $C_2 = g^{v_2} h^{r_2} \in \mathbb{G}$. To build an encryption of product $v_1 \cdot v_2 \bmod n$ 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

$$\begin{aligned} C &= e(C_1, C_2) h_1^r = e(g^{v_1} h^{r_1}, g^{v_2} h^{r_2}) h_1^r \\ &= g_1^{v_1 v_2} h_1^{v_1 r_2 + v_2 r_1 + q_2 r_1 r_2 \alpha + r} = g_1^{v_1 v_2} h_1^{r'} \in \mathbb{G}_1 \end{aligned}$$

where $r' = v_1 r_2 + v_2 r_1 + q_2 r_1 r_2 \alpha + r$ is distributed uniformly in $\mathbb{Z}_n$ as required. Thus, C is a uniformly distributed encryption of $v_1 v_2 \bmod n$, 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 v in $\mathbb{G}$ as $E_G(v)$ and one in $\mathbb{G}_1$ as $E_{G_1}(v)$.

2.4 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 and the shares of the corresponding private exponent. At the end of the computation, N becomes public and the players are convinced that N is a product of two large primes by using zero-knowledge proof. Then, following the algorithm introduced 2.3.2, Bilinear group $(\mathbb{G}, \mathbb{G}_1, e)$ is also generated from N. 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 the players to jointly decrypt the ciphertexts.

3. New Efficient Auction Protocol

In this section, we show bit-slice auction protocols based on the evaluation of multivariate polynomials with the total degree of two on encrypted values. Both first and second price auction protocols on 2-DNF formula on encrypted values and with the mix-and-match protocol. To maintain secrecy of the players' bidding prices 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 [9]. Here, we define three types of new tables, $Select_k$, MAP_1 and MAP_2 for the second price auction. In the proposed protocol, the MAP_1 and MAP_2 tables are created among AM before an auction. On the other hand, $Select_k$ is created through the protocol corresponding to the players' inputs. The AM jointly computes values in the mix-and-match table for distributed decryption of plaintext equality test. Table $Select_k$ is also used for the second price auction protocol in [9]; MAP_1 and MAP_2 are new tables that we propose. Given a message t, MAP_1 and MAP_2 are tables for mapping an encrypted value $a_1 \in E_{G_1}(t)$ (which is an output of a computation with one multiplication) to $a_2 \in E_G(t)$. Table $Select_k$ has $2k + 1$ input bits and k output bits as follows.

$$Select_k(b, x_{k-1}, \ldots, x_0, y_{k-1}, \ldots, y_0) = \begin{cases} (x_{k-1}, \ldots, x_0) & if\ b = 1 \\ (y_{k-1}, \ldots, y_0) & otherwise \end{cases}$$

For two encrypted input vectors $(x_{k-1}, \ldots, x_0)$ and $(y_{k-1}, \ldots, y_0)$, b is an encryption of the check bit that selects which vector to output, $(x_{k-1}, \ldots, x_0)$ or $(y_{k-1}, \ldots, y_0)$. For secure computation, the AM re-encrypts the output vector. In the proposed protocol, the $Select_k$ table is created through the auction to update W corresponding to an input value $E(b_j)$. 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 $x_1 \in \{E_{G_1}(0), E_{G_1}(1), \ldots, E_{G_1}(m)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$. These tables can be constructed using the mix-and-match protocol because the Boneh-Goh-Nissim encryption has homomorphic properties.

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

3.1 First Price Auction Using 2-DNF Scheme and Mix-and-Match Protocol

We assume m players, $P_1, \ldots, P_m$ and a set of auction managers, AM. We can assume that AM is either a subset of players or a different group such as management group for auctions. The players bid their encrypted prices. Then, the AM executes an auction with players' encrypted bids and at the end of the auction and jointly decrypts the results of the protocol. Players find the highest price through the protocol and the winner by decrypting the results.

3.1.1 Setting

AM jointly generates and shares private keys among themselves 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

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

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, $r_1, r_2 \in \mathbb{Z}_n^*$ are random numbers and $c'_1 \in E_{G_1}(b_1)$ and $c'_2 \in E_{G_1}(b_2)$. We define two polynomial time computable operations Mul and $\otimes$ by applying a 2DNF formula for AND, OR respectively.

$$Mul(c_1, c_2) = e(c_1, c_2) = e(g^{b_1}h^{r_1}, g^{b_2}h^{r_2}) \in E_{G_1}(b_1 \wedge b_2)$$
$$c'_1 \otimes c'_2 \in E_{G_1}(b_1 + b_2)$$

by applying a 2DNF formula for AND.

The AM generates $W = (w_1, \ldots, w_m)$, where each $w_j = 1$, and encrypts them as $\widetilde{W} = (\tilde{w}_1, \ldots, \tilde{w}_m)$. The 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 $\widetilde{W} = (\tilde{w}_1, \ldots, \tilde{w}_m)$, AM computes $s_{i,j} = Mul(\tilde{w}_i, c_{i,j})$ for each player i, and

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

(Step 1-b) The AM uses table MAP_1 for $s_{i,j}$ for each i and finds the values of $\tilde{s}_{i,j}$. Let $\widetilde{S}_j = (\tilde{s}_{1,j}, \ldots, \tilde{s}_{m,j})$. The 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 $b_{max}^{(j)} = 1$, AM update $\widetilde{W} = \widetilde{S}_j$, otherwise $\widetilde{W}$ remains.

(Step 2) For the final $\widetilde{W} = (\tilde{w}_1, \ldots, \tilde{w}_m)$, AM decrypts each $\tilde{w}_i$ with the verification protocols and obtains plaintext w_i. The highest price is obtained as $B_{max} = (b_{max}^{(k-1)}, \ldots, b_{max}^{(0)})_2$. P_i is a winner if and only if $w_i = 1$.

3.2 Second Price Auction Using 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. The setting and bidding phases are the same as those for 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 $\tilde{w}_j \in E_G(1)$ shown above.

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

(Step 1-a) For $\widetilde{W} = (\tilde{w}_1, \ldots, \tilde{w}_m)$, AM computes $s_{i,j} = Mul(\tilde{w}_i, c_{i,j})$ for each player i, and

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

(Step 1-b) The AM uses table MAP_1 for $s_{i,j}$ for each i and finds the values of $\tilde{s}_{i,j}$. Let $\widetilde{S}_j = (\tilde{s}_{1,j}, \ldots, \tilde{s}_{m,j})$. The AM also uses the table MAP_2 for h_j as an input value. By using this table, AM retrieves $E(b_j) \in E_G(0)$ if h_j is a ciphertext of 0, otherwise he retrieves $E(b_j) \in E_G(1)$.

(Step 1-c) AM creates the table $Select_k$ as input values $(E(b_j), \widetilde{S}_j, \widetilde{W})$.

The AM executes $\widetilde{W} = Select_k(E(b_j), \widetilde{S}_j, \widetilde{W})$, that is, if $E(b_j)$ is the encryption of 1, $\widetilde{W}$ is updated as $\widetilde{S}_j$.

(Step 2) For the final $\widetilde{W} = (\tilde{w}_1, \ldots, \tilde{w}_m)$, AM decrypts each $\tilde{w}_i$ with verification protocols and obtains the plaintext w_i. P_i is the winner if and only if $w_i = 1$. The AM remove the player who bids the highest price and run the first price auction protocol again. The second highest price is obtained as $B_{max} = (b_{max}^{(k-1)}, \ldots, b_{max}^{(0)})_2$.

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 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 $x \in Z_n^*$ 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 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 the winner and the highest price or the second highest price (the first price auction and second price auction respectively). An auction scheme is secure if there is no polynomial time adversary that breaks privacy with non-negligible advantage $\epsilon(\tau)$. Here, we introduce the mix-and-match oracle which enable to execute an auction protocol without calculation of plaintext equality test and mix-and-match. The calculation of plaintext equality test and mix-and-match are used in proposed auction protocols such as checking whether h_j is 0 and updating $\widetilde{W}$. However, to calculate plaintext equality test and mix-and-match needs the secret key of auction protocols. The reason we put mix-and-match oracle in security proof is to execute auction protocols without the secret key. Then, 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. An adversary can use the mix-and-match oracle which receives an encrypted value $x_1 \in E_{G_1}(m)$ and returns the encrypted value $x_2 \in E_G(m)$ for given a message m according to the mix-and-match table shown in Table 3 (which has the same function as MAP_2). We consider cases only where the range of input values is $\{0,1,\ldots,m\}$ and the range of the output is $\{0,1\}$ for MAP_2. MAP_1 can also be computed if the range of the input value is restricted in $\{0,1\}$. If mix-and-match oracle receives an improper value, the request is supposed to be aborted. An adversary can also calculate $Select_k(b, x_{k-1}, \ldots, x_0, y_{k-1}, \ldots, y_0) = b(x_{k-1}, \ldots, x_0) + (1-b)(y_{k-1}, \ldots, y_0)$ with an additional polynomial computation. Because, by 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. 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 $\mathcal{B}$.

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

Fig. 1 $EXPT_{A,\Pi}$.

Definition 1:
Let $\Pi = (KeyGen, Encrypt, Decrypt)$ 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.

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

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

Definition 2: Let Π = (*KeyGen*, *Bid*, *WinnerDecision*) be a secure auction protocol, and let B be two probabilistic polynomial-time algorithm B_1 and B_2.

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

where $EXPT_{B,\Pi}$ is a semantic security game of the privacy of the auction protocol shown in Fig. 2. *Bid* is the function of encrypting the bidding price of each player. *WinnerDecision* is the function of executing the auction with encrypted bids in order to find the winner and winning price. First of all, B_1 generates k-bit integers, $b_1, b_2, \ldots, b_{m-1}$ as plaintexts of bidding prices for player $P_1,\ldots,P_{m-1}$, 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 first price bid in a first price auction and the second highest price in a second price auction. Then the function Bid is used for encrypting players' bidding prices such as $(c_1 = Bid(PK, b_1), c_2 = Bid(PK, b_2), \ldots, c_{m-1} = Bid(PK, b_{m-1}), c_m = Bid(PK, b_{m_b}))$ where $b \xleftarrow{r} \{0,1\}$. Finally the auction is executed with the function $WinnerDecision(c_1, c_2, \ldots, c_{m-1}, c_m)$ as the players' encrypted bidding prices. 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.

(Proof) We show if there is adversary $\mathcal{B}$ that breaks the security of the proposed auction protocol, we can

$$(PK, SK) \leftarrow KeyGen$$
$$(b_1, b_2, \ldots, b_{m-1}, b_{m_0}, b_{m_1}, s) \leftarrow B_1(PK)$$
$$b \leftarrow \{0, 1\}$$
$$c_1 \leftarrow Bid(PK, b_1), c_2 \leftarrow Bid(PK, b_2), \ldots, c_{m-1} \leftarrow Bid(PK, b_{m-1}), c_m \leftarrow Bid(PK, b_{m_b})$$
$$(winner, winning\ price) \leftarrow WinnerDecision(c_1, c_2, \ldots, c_{m-1}, c_m)$$
$$b' \leftarrow B_2(winner, winning\ price, s, view_{WinnerDecision})$$
$$return\ 1 \text{ iff } b = b'$$

Fig. 2 $EXPT_{B,\Pi}$.

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 $Encrypt(PK, m_b^i)$ and executes a secure auction protocol with the mix-and-match oracle. When calculation of plain equality test or mix-and-match is needed such as checking whether h_j is 0 and updating $\widetilde{W}$, $\mathcal{A}$ uses mix-and-match oracle to transfer encrypted value over E_{G_1} to E_G. b_{m_0} and b_{m_1} are not the winning bidding prices and $\mathcal{A}$ knows all the input values, $b_1, b_2, \ldots, b_{m-1}$ except the i-th bit of b_{m_b}. So, $\mathcal{A}$ with mix-and-match oracle can simulate an auction for the adversary of auction $\mathcal{B}$. Through the auction, $\mathcal{B}$ observes the calculation of the encrypted values 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 Property 1 introduced in Sect. 2.2, 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. Comparison of Auction Protocols

4.1 First Price Auction

BGN encryption is based on bilinear groups of composite order. So, we assume the keys size of N is 1024 bit to be infeasible to factor. The protocol proposed in [9] requires mk AND computations to calculate $S_j = (Mul(\tilde{w}_1, c_{1,j}), \ldots, Mul(\tilde{w}_m, c_{m,j}))$ for $j = k$ -1 to 0 and k plaintext equality tests when it checks whether $b_{max}^{(i)}$ is the ciphertext of 0. One AND computation requires two plaintext equality tests. So, the total number of plaintext equality tests is $2mk + k$. On the other hand, in the proposed protocol, S_j can be computed by addition and multiplication of ciphertexts with 2-DNF scheme. It also requires mk MAP_1 computation to update $\widetilde{W}$, k plaintext equality tests to check $b_{max}^{(i)}$ and mk pairing calculation. A comparison between the proposed protocol and that in [9] is shown in Table 4. Although the number of PET in the proposed protocol is reduced compared to the protocol in [9], the proposed protocol needs mk pairing calculation. The computation cost of pairing calculation is approximately 4 times than that of group calculation in the worst case [8]. Thus, the complexity of one pairing is much less than that of one PET. Therefore, for the evaluation of efficiency, the greatest factor is the number of PET and the proposed protocol for first price auction is certainly more efficient than that in [9]. As for the communication costs, communication during Bidding and Opening phase in [9] and proposed protocol is the same, so it depends on the encrypted message sizes(that is, proportional to the key sizes) of each protocol.

Table 4 Comparision in the first price auction.

	Proposed	[9]
Type of Encryption	Elliptic Curve	Finite Field
Modular	mod $n(= p \times q)$	mod p
PET(Total)	$mk + k$	$2mk + k$
Pairing	mk	0

4.2 Second Price Auction

In the second price auction protocol, the protocol in [9] requires $(2m - 1)k$ AND, $(m - 2)k$ OR and k $Select_k$ gates. One OR gate requires two plaintext equality tests. $Select_k$ requires one test to check whether b is the ciphertext of 1, so in total approximately $6mk - 5k$ plaintext equality tests are required. Conversely, the proposed protocol requires MAP_1 $2mk$ times and MAP_2 k times. MAP_1 requires one plaintext equality test which uses to check whether input value is a ciphertext of 0 or 1. The range of input value in the table MAP_2 is $m + 1$ (from 0 to m) and use one plaintext equality test for each column in the mix-and-match table. MAP_2 requires approximately $m/2$+1 times on average. It also requires k plaintext equality tests to decide the second highest price among the rest of player except the winner. In total, the calculation cost for PET is $5/2mk + 2k$. mk and $(m - 1)k$ times pairing calculation are needed to decide the winner of auction and the second highest price respectively. A comparison between the proposed protocol and that in [9] is shown in Table 5. In the second price auction we can re-

Table 5 Comparision in the second price auction.

	Proposed	[9]
Type of Encryption	Elliptic Curve (1024 bit)	Finite Field (1024 bit)
Modular	mod $n(= p \times q)$	mod p
PET(Total)	$5/2mk + 2k$	$6mk - 5k$
Pairing	$2mk - k$	0

duce the number of times when the plaintext equality test is executed.

5. Conclusion

We introduced new efficient auction protocols based on the BGN encryption and showed that they are more efficient than that proposed in [9]. 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," Proc. Public Key Cryptography 2002, LNCS vol.2274, pp.115–124, 2002.
[2] D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," Invited paper Public Key Cryptography 1998, LNCS, vol.1431, pp.1–13, 1998.
[3] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," Proc. Theory of Cryptography (TCC) 2005, LNCS, vol.3378, pp.325–341, 2005.
[4] D. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms," Commun. ACM, ACM 1981, pp.84–88, 1981.
[5] M.K. Franklin and M.K. Reiter, "The design and implementation of a secure auction service," IEEE Trans. Softw. Eng., vol.22, no.5, pp.302–312, 1995.
[6] M. Jakobsson and A. Juels, "Mix and match: Secure function evaluation via ciphertexts," Proc, Asiacrypt 2000, LNCS, vol.1976, pp.162–177, 2000.
[7] A. Juels and M. Szydlo, "A two-server sealed-bid auction protocol," Proc. Financial Cryptography 2002, LNCS, vol.2357, pp.72–86, 2002.
[8] T. Kerins, W.P. Marnane, E.M. Popovici, and P.S.L.M. Barreto, "Hardware accelerators for pairing based cryptosystems" IEE Proc. Information Security 2005, vol.152, no.1, pp.47–56, 2005.
[9] K. Kurosawa and W. Ogata, "Bit-slice auction circuit," Proc. 7th European Symposium on Research in Computer Security 2002, LNCS, vol.2502, pp.24–38, 2002.
[10] H. Lipmaa, N. Asokan, and V. Niemi, "Secure Vickrey auctions without threshold trust," Proc. 6th Annual Conference on Financial Cryptography, LNCS, vol.2357, pp.87–101, 2002.
[11] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design" Proc. 1st ACM Conference on Electronic Commerce (ACM-EC), pp.129–139, ACM press 1999, 1999.
[12] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," Proc. Eurocrypt 1998, LNCS, vol.1403, pp.308–318, 1998.
[13] P. Pallier, "Public-key cryptosystems based on composite degree residuosity classes," Proc. Eurocrypt 1999, LNCS, vol.1592, pp.223–238, 1999.
[14] C. Park, K. Itoh, and K. Kurosawa, "All/nothing election scheme and anonymous channel," Proc. Eurocrypt 1993, LNCS, vol.765, pp.248–259, 1993.
[15] Y. Tamura, T. Shiotsuki, and A. Miyaji, "Efficient proxy-bidding system," IEICE Trans. Fundamentals (Japanese Edition), vol.J87-A, no.6, pp.835–842, June 2004.

Takuho Mitsunaga received the B.Ec. and M.E. degrees from Osaka University and Kyoto University, Osaka and Kyoto, Japan, in 2008 and 2010, respectively. Currently, he is a doctor course student of Kyoto University. His research interests are cryptograpy and information security

Yoshifumi Manabe received the B.E., M.E., and Dr.E. degrees from Osaka University, Osaka, Japan, in 1983, 1985, and 1993, respectively. In 1985, he joined Nippon Telegraph and Telephone Corporation. Currently, he is a senior research scientist, supervisor of NTT Communication Science Laboratories. His research interests include distributed algorithms, cryptography, and graph theory. He has been a guest associate professor of Kyoto University since 2001. He is a member of ACM, IPSJ, JSIAM, and IEEE.

Tatsuaki Okamoto received the B.E., M.E., and Dr.E. degrees from the University of Tokyo, Tokyo, Japan, in 1976, 1978, and 1988, respectively. He is a Fellow of NTT Secure Platform Laboratories. He is presently engaged in research on cryptography and information security. Dr. Okamoto is a guest professor of Kyoto University.