IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

VOL. E99-A NO. 8
AUGUST 2016

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

A Secure M + 1st Price Auction Protocol Based on Bit Slice Circuits*

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

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

key words: *secure auction protocol, M+1st 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 [18]. 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 $M + 1st$ price auction. 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 prices higher than the winning price are the winning bidders, and each winning bidder buys one unit of the goods at the winning price. A simple solution to construct a secure auction protocol 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 [19] 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 [9].

Boneh, Goh and Nissim suggested a public evaluation system for polynomials of a total degree of two on encrypted values named 2-DNF [3]. Their scheme has additive homomorphism in addition to the bilinear map, which allows one multiplication on encrypted values. This property in turn allows the evaluation of multivariate polynomials of a total degree of two on encrypted values.

Mitsunaga, Manabe and Okamoto proposed an efficient first and second price auction protocols based on Boneh-Goh-Nissim encryption [3].

In this paper, we introduce an efficient secure auction protocol for $M + 1st$ 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 [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. Kurosawa and Ogata have proposed an efficient verifiable auction protocol for the

Manuscript received November 30, 2015.
Manuscript revised May 2, 2016.

[†]The author is with JPCERT/CC, Tokyo, 101-0054 Japan.
[††]The author is with Kogakuin University, Tokyo, 163-8677 Japan.
[†††]The author is with NTT Corporation, Musashino-shi, 180-8585 Japan.

*Preliminary version of this paper was presented at IWSEC 2011 [14]. This paper is based on [14] but containing some additional explanation of the proposed protocol in Section 3.1, 3.3, 3.4 and 4. Most significantly, Theorem 1 is added to explain the correctness of the proposed protocol.

a) E-mail: mitsunaga@ai.soc.i.kyoto-u.ac.jp
b) E-mail: manabe@cc.kogakuin.ac.jp
c) E-mail: okamoto.tatsuaki@lab.ntt.co.jp

DOI: 10.1587/transfun.E99.A.1591

first and second price auction with bit-slice approach [9]. To keep the privacy of players bidding prices, they introduced mix-and-match protocol to their protocol. Players' bidding prices are represented as binary expression, therefore the complexity of each player is proportional to log p, for potential bidding price p. Mitsunaga, Manabe and Okamoto suggested secure auction protocols for the first and second price auction. They applied Boneh-Goh-Nissim Encryption to the bit-slice auction protocol to improve computation costs [13].

For $M+1st$ price auction, Lipmaa, Asokan and Niemi proposed an efficient secure $M+1st$ auction scheme [10]. In their 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]. However in their scheme, a player's bid is not a binary expression. Thus, its time complexity is $O(m\log p)$ for a m-player and p-bidding price auction.

1.3 Our Result

This paper presents an efficient secure auction protocol for $M+1st$ price auction. By adding bits of players' statuses on "bit-slice auction" in [9], we expand the first and second price auction protocol to $M+1st$ price auction protocol. In our proposed protocol, bidding prices are represented as binary expression. Thus, when 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 [1], the complexity is a polynomial of p and m. Our proposed protocol is a generalized protocol of first and second price auction [13]. However, in case of second price auction ($M=1$), the proposed protocol is approximately twice faster than the one in [13]. Since the protocol in [13] needs to execute auction protocol twice, one for deciding the winner and the other for deciding the winning price, while in the proposed protocol both the winner and winning price can be decided by executing auction protocol once.

2. Preliminaries

2.1 The Model of Auction and Outline of Auction Protocol

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. For each player P_i's bidding price Z_i, the aim of the protocol is for players to compute $f(Z_1, \ldots, Z_m)$ without revealing any additional information. Its outline is as follows.

1. **Input stage:** Each $P_i (1 \le i \le m)$ computes ciphertexts of the bits of Z_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 evaluate each gate, G_j in C_f, in order.
3. **Output stage:** After evaluating the last gate G_M, the players obtain O_M, a ciphertext encrypting $f(Z_1, \ldots, Z_m)$. They jointly decrypt this ciphertext value to reveal the output of function f.

2.2 Mix and Match Protocol

2.2.1 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.

2.2.2 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 and ξ'_i are lists of l ciphertexts, $(c_1, \ldots, c_l)$, for some $l \ge 1$. For all players to verify the validity of $(\xi'_1, \ldots, \xi'_L)$, we use the universal verifiable MIX net protocol described in [17].

2.2.3 Plaintext Equality Test (PET)

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

Table 1 Mix-and-match table for AND.

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

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

There is a case where PET fails with negligible probability even if $v_1 \neq v_2$. In that case it is hard to detect the failure in PET scheme.

2.2.4 Mix and Match Stage

For each logical gate, $G(x_1, x_2)$, of a given circuit, m players jointly compute $E(G(x_1, x_2))$ 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 $c'_i \in E(x_1 \wedge x_2)$.

2.3 Evaluating 2-DNF Formulas on Ciphertexts

Given encrypted Boolean variables $x_1, \ldots, x_m \in \{0, 1\}$, 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

The construction in [3] 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 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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ 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 introduce the public key system which resembles the Pallier [16] and the Okamoto-Uchiyama encryption schemes [15]. 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 $u \overset{R}{\leftarrow} \mathbb{G}$ 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 $\{0, 1, ..., T\}$ 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 $r \in \mathbb{Z}_n$ 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 $v_1 + v_2$ mod n by computing the product $C = C_1 C_2 h^r$ for a random number $r \in \mathbb{Z}_n$. 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 $v_1 \cdot v_2$ mod 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

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

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 $v_1 v_2$ mod 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 jointly generate 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, 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 auction managers. We can assume that auction managers are either a subset of players or a different group such as management group for auctions.

3. New Efficient Auction Protocol

In this section, we show an efficient $M + 1st$ price auction based on bit-slice auction protocols. Compared to previous works on secure $M + 1st$ price auctions, the proposed protocol is more efficient because bidding prices are represented as binary numbers. However if a quite large number of players participate in an auction, it still needs high computation costs, because the complexity of proposed protocol is a polynomial of m for the m-player auction. If some players bid the same price which is more than M highest price, such as a case 2 players bid the same price as 3rd highest price for 5-player auction for 3 goods, this protocol does not work well. (Regarding to this situation called Tie-Break, see [12] for more details.) At the end of auction, winners and winning price can not be decided.

3.1 Proposed $M + 1st$ Price Auction Protocol

We show how to find the winners and the winning bidding price with unencrypted bidding prices. Through an auction, players are labeled as three types of players' statuses, winner(s), candidate(s) and survivor(s) described as follow.

- $Winner$: a player who decided to be a winner.
- $Candidate$: a player who is not decided to be a winner but has a possibility of $M + 1st$ highest bidder.
- $Survivor$: a candidate on the current and his bid on the bit is 1.

This auction protocol starts from the highest bit of players' bidding prices and proceeds to lower one bit by one bit. At the beginning of the auction, all players are Candidates since no player is decided as a Winner and all players have possibilities to win the auction. On each bit, a status of a player is decided by comparing players' bidding prices. If a player's bidding price is found to be larger than $M + 1st$ highest bit, his status becomes a Winner. On the other hand, the bidding prices is found to be smaller than $M + 1st$ highest bit, he loses a status of a Candidate, because he no longer has a possibility to win the auction. Otherwise, while he has a chance to be a Winner or $M + 1st$ highest bidder, he keeps his status a Candidate. At the end of the auction, the winners and the winning price is found according to the players' bidding prices. To explain precisely, we also define the players in the variables of winner(s), candidate(s) and survivor(s) on j-th bid as W_j, C_j and S_j respectively and the numbers of elements whose value is 1 in W_j and S_j as $|W_j|$ and $|S_j|$.

- $W_j[1 \ldots m]$: $W_j[i]$=1 if player P_i is decided to be a winner by upper $k - j$ bits of the bid.
- $C_j[1 \ldots m]$: $C_j[i]$=1 if player P_i is not decided to be a winner but has a possibility of $M + 1st$ highest bidder by upper $k - j$ bits of the bid.
- $S_j[1 \ldots m]$: $S_j[i]$=1 if player P_i is a candidate on j-th bit ($C_j[i]$=1) and his bid on j-th bit is 1.

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

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

For $j = k - 1$ to 0

$S_j[i] = C_{j+1}[i] \wedge z_i^{(j)}$ $(1 \le i \le m)$

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

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

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

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

Table 2 Example of 5-player auction for 3 goods.

	$j=4$		$j=3$				$j=2$				$j=1$				$j=0$			
	C_4	W_4	K_3	S_3	C_3	W_3	K_2	S_2	C_2	W_2	K_1	S_1	C_1	W_1	K_0	S_0	C_0	W_0
$Z_1 = (1011)_2$	1	0	1	1	0	1	0	0	0	1	1	0	0	1	1	0	0	1
$Z_2 = (0111)_2$	1	0	0	0	1	0	1	1	1	0	1	1	0	1	1	0	0	1
$Z_3 = (0101)_2$	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1
$Z_4 = (0100)_2$	1	0	0	0	1	0	1	1	1	0	0	0	1	0	0	0	1	0
$Z_5 = (0001)_2$	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
$\|W\|$ and $\|S\|$		0		1		1		3		1		1		2		1		3
Z_{M+1st}				0				1				0				0		

else

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

$C_j[i] = C_{j+1}[i] \wedge \overline{S_j[i]}$ $(1 \le i \le m)$

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

end

end

For a player P_i $(1 \le i \le m)$, if P_i is decided to be a winner by j-th bit from high-order bits of the bid, then $W_j[i] = 1$. If player P_i is not decided to be a winner but has a possibility of $M+1st$ highest bidder on the j-th bit, then $C_j[i]$=1. If $C_j[i]$=1 and j-th bit of P_i's bid is 1, then $S_j[i] = 1$.

If the number of Winners on $(j+1)$-th bit and Survivors on j-th bit is more than or equal to $M+1$, we keep Winners remained and update players' status Candidates to eliminate players i whose bidding prices are 0 on this bit. If the number of Winners on $(j+1)$-th bit and Survivors on j-th bit is less than $M+1$, Survivors on j-th bid are determined as Winners, so we update W_j as $W_{j+1}[i] \vee S_j[i]$ and eliminate player i that satisfies $S_j[i]$=1.

Theorem 1. *In the above algorithm,*

- *For the vector W_0, P_i is the winner of the auction if and only if $W_0[i] = 1$.*
- *Z_{M+1st} is the M+1st bidding price.*

Proof. We show the values of Winners, Candidates and Survivors satisfy the definition for all l bits by induction and the winning price, Z_{M+1st}, is consistent with the bidding prices of players.

We show that the variables satisfy the definitions through the proposed auction protocol by induction. In this proof, we denote the $M+1st$ bidding players by P_{M+1st}.

- Initial Step:

When $l = k$, following the initial setting, Winner is a null vector, and the statuses of all players are Candidate. $z^{(1)}_{M+1st}$ is a blank(not defined). Thus this situation satisfies the definition of the players statuses.

- Inductive step:

When $l = j+1$ we assume the definition of each player status holds on $(j+1)$-th and upper bits, then we show that the definition of each player status holds when $l = j$ that is;

(1). If the number of Winners by the upper $(j+1)$-th bits of Z_i and Survivors on j-th bit is more than or equal to $M+1(|W_{j+1}| + |S_j| \ge M+1)$, new Winners can not selected on this bit, because if Survivors become Winners, the number of Winners exceeds the number of goods M. The players in the status of Winners do not change. Survivors (Candidates whose bids on this bit are 1) become Candidates of next bit because they have a chance to be a Winner or P_{M+1st}. The rest of Candidates$(C_{j+1} - S_j)$ lose the auction since their bidding prices are found to be smaller than Z_{M+1st}. Thus, the definition of players' status holds. In this case, Z_{M+1st} is bigger than or equal to the lowest bid of Survivors, which is the $(|W_{j+1}| + |S_j|)$-th highest bid, then P_{M+1st} is categorized as a Survivor. Thus $Z^{(j)}_{M+1st}$ is 1.

(2). If the number of Winners by the upper $(j+1)$-th bits and Survivors on j-th bit is is less than $M+1(|W_{j+1}| + |S_j| < M+1)$, Survivors are decided to be Winners, since their bidding prices are found to be larger than Z_{M+1st}. On the other hands, the players in $C_{j+1} - S_j$ become Candidates, since they still have a chance to be a Winner or P_{M+1st}. Thereby showing that in the both situation the definition of each player status holds when $l = j$. In this case, Z_{M+1st} is smaller than the $(|W_{j+1}| + |S_j|)$-th highest bid and P_{M+1st} is in the group of $C_{j+1} - S_j$. Thus, $Z^{(j)}_{M+1st}$ is 0.

□

3.2 Example

We show an example of 5-player auction for 3 goods $(M = 3)$ in Table 2. The information we need to find is the first, second and third highest bidders as the winners of the auction and the forth highest bidding price as the winning price. Assume each player's bid as follows,

$Z_1 = (1011)_2 = 11$

$Z_2 = (0111)_2 = 7$

$Z_3 = (0101)_2 = 5$

$Z_4 = (0100)_2 = 4$

$Z_5 = (0001)_2 = 1$

So, the winners are P_1, P_2 and P_3 and the winning price is $Z_4 = (0100)_2 = 4$. In Table 2, we denote by K_j the vector comprising the $k-j$-th MSB of each player's bid.

For initial setting $j = 4$, all players are Candidates, since all players have possibilities to win the auction according to the definition of the player status. They are not decided to win the auction yet, so none of players' statuses is Winners.

Next step $j = 3$, only z^4_1 is 1, so P_1 is decided to be Survivor and the number of Winner on upper bit and Survivor on 4th bid is 1. Then, by following the protocol, P_1 becomes Winner and is removed from Candidate. The other players are kept to be Candidates to compete the auction. Next step $j = 2$, bids of Z_2, Z_3 and Z_4 are 1, so they are decided as

Survivors. The number of Winner on upper bit and Survivor on 3rd bid is 4, which means P_2, P_3 and P_4 can not decided to be Winners but kept to be Candidates and P_5 already loses the auction. Following the protocol, from the 1st bits of the bids P_1, P_2 and P_3 are decided to be Winners. The winning price $Z_4 = (0100)_2$ is shown in the row of Z_{M+1st} in the Table 2.

3.3 Secure $M + 1st$ 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. The players bid their encrypted prices and broadcast them. The AM runs an auction protocol with the encrypted bids and after the auction AM jointly decrypts the results of the protocol and broadcast it to the players. Players can verify the winning price (the $M + 1st$ price) and the winners from the encrypted bidding prices by using verification protocols. To maintain secrecy of the players' bidding prices through the protocol, we need to use the mix-and-match protocol. Here, we define two 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 is used for transferring encrypted values of 0 and 1 in $\mathbb{G}_1$ to encrypted values of 0 and 1 in $\mathbb{G}$ respectively. This mapping, $x_1 \in \{E_{G_1}(0), E_{G_1}(1)\} \mapsto x_2 \in \{E_G(0), E_G(1)\}$, is shown in Table 3. The table MAP_2 is a function for mapping $x_1 \in \{E_{G_1}(0), E_{G_1}(1), \ldots, E_{G_1}(m)\} \mapsto x_2 \in \{E_G(0), E_G(1)\}$. This is used for transferring encrypted values of {0,...,M} and M + 1,...,m} in $\mathbb{G}_1$ to encrypted values of 0 and 1 in $\mathbb{G}$, respectively as described in Table 4. These tables can be constructed using the mix-and-match protocol because the Boneh-Goh-Nissim encryption has homomorphic properties.

3.3.1 Setting

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

3.3.2 Bidding Phase

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

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

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].

Table 3 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 4 Table for MAP_2.

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

3.3.3 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 by applying a 2-DNF formula for AND, and $\otimes$ by the operation of addition.

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

AM executes PET for MAP_1 and MAP_2 in this open phase to keep the secrecy of players bidding prices through the auction. Let $C_k = (c_1^k, \ldots, c_m^k)$, where each $c_i^k \in E_G(1)$ and $W_k = (w_1^k, \ldots, w_m^k)$, 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 $C_j = (c_1^j, \ldots, c_m^j)$, AM computes $s_i^j = Mul(c_i^{j+1}, b_i^j)$ for each player i, and

$$S_j = (s_1^j, \ldots, s_m^j) = (Mul(c_1^{j+1}, b_1^j), \ldots, Mul(c_m^{j+1}, b_m^j))$$
$$h_j = Mul(c_1^{j+1}, b_1^j) \otimes \cdots \otimes Mul(c_m^{j+1}, b_m^j)$$
$$d_j = w_1^j \otimes \cdots \otimes w_m^j$$

(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 $\widetilde{S}_j = (\tilde{s}_1^j, \ldots, \tilde{s}_m^j)$.
(Step 1-c) AM uses table MAP_2 for $d_j \otimes h_j$ and decrypts the output value. The reason MAP_2 is used here is to prevent AM finding any other information except $d_j \otimes h_j$ is more than $M + 1$ or not. If the output value is 1, the number of winners and survivors are more than or equal to $M+1$. Then, AM updates

$W_j = W_{j+1} = (w_1^{j+1}, \ldots, w_m^{j+1})$
$C_j = \widetilde{S}_j = (\tilde{s}_1^j, \ldots, \tilde{s}_m^j)$
$z_{M+1st}^{(j)} = 1$

If the output value is 0, then

$W_j = W_{j+1} + S_j = (w_1^{j+1} \otimes s_1^j, \ldots, w_m^{j+1} \otimes s_m^j)$
$C_j = C_{j+1} - \widetilde{S}_j = (c_1^{j+1} \otimes (\tilde{s}_1^j)^{-1}, \ldots, c_m^{j+1} \otimes (\tilde{s}_m^j)^{-1})$
$z_{M+1st}^{(j)} = 0$

There is no case where $C_{j+1}[i] = 0$ and $\widetilde{S}_j[i] = 1$ for all

players ($1 \leq i \leq m$). Thus $C_{j+1}[i] - \widetilde{S}_j[i]$ can be properly calculated.

(Step 2) For the final $W_0 = (w_1^0, \ldots, w_m^0)$, 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 plaintext of $w_i^0 = 1$ and $\sum_{i=1}^{m} w_i^0 = M$. The $M+1st$ highest price is obtained as $Z_{M+1st} = (z_{M+1st}^{(k-1)}, \ldots, z_{M+1st}^{(0)})_2$.

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, 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 as an inverse of the ciphertext C($f = C(g^b)^{-1}$). 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.

3.4 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 4. (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,. . . ,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 semantically 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 $\mathcal{B}$.

$$
\begin{array}{c}
(PK, SK) \leftarrow KeyGen \\
(m_0, m_1, s) \leftarrow A_1^{O_1}(PK) \\
b \leftarrow \{0,1\} \\
c \leftarrow Encrypt(PK, m_b) \\
b' \leftarrow A_2^{O_1}(c, s) \\
return\ 1 \text{ iff } b = b'
\end{array}
$$

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

Definition 1.
Let Π *= (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.

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

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.

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

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 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 $M+1st$ highest price. 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 2. *The privacy of the auction protocols is*

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

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

Table 5 The comparison of computational complexity.

	[1]	Proposed
Bidding(per one bidder)	p encryptions	log p encryptions
Running auction (Calculation over group)	$2mp$ multiplications	$m$log p multiplications $m$log p pairing
Running auction(PET)	log $p(M+1)$ times	log $p(M+1)$ + mp times
Decrypting to decide the winners	m decryptions	m decryptions
Decrypting to decide the winning price	log p decryptions	log p decryptions

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 compose adversary $\mathcal{A}$ that breaks the semantic security of the BGN encryption with the mix-and-match oracle. $\mathcal{B}$ generates k-bit integers, $b_1, b_2, \ldots, b_{m-1}$ 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 following the definition. $\mathcal{A}$ 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 c as a result of $Encrypt(PK, m_b^i)$ and send it to $\mathcal{B}$. $\mathcal{B}$ receives $c_1, \ldots, c_{m-1}$, and c as c_m as the result of function *Bid* and uses *WinnerDecision* function to execute 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 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 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. There is a case where PET fails with negligible probability as described in 2.2.3. However, the failure of PET brings the miscalculation of auction result. For example, if PET used for the transformation of s_j^i fails, it brings a false winner or loser. We assume that AM proceeds the auction properly with verification protocol, thus in that kind of case players can detect the failure of PET with verification protocol.

4. Comparison of Auction Protocols

The protocol proposed in [1] is based on homomorphic encryption. In their protocol, each player encrypts his bidding price is not represented as binary bit. Therefore, for a potential bidding price p and m players, each player needs to execute encryption p times for bidding, and AM calculates multiplications of ciphertexts $2mp$ times to run the auction. PET (plaintext equality test) is used in the opening phase to check whether the number of i-th bid is more than $M+1$ or not with using binary search for each price i in $[1, p]$. Binary search for p needs log p comparisons and one comparison needs PET $M+1$ times for each bid to check whether it is more than $M+1$. In the end of auction, m and log p decryptions are used to decide the winner and winning price of the auction.

Our auction protocol is based on BGN encryption where each player's bidding price is represented as a binary expression. We use PET mp times when AM calculates $\tilde{s}_i^j$ from player j's i-th bid for all i and j. We also use PET when AM detects whether $b_{M+1st}^{(i)}$ is more than M or not. log p decryptions are used to open the winning price and m decryptions are used to open the winners of auction. A comparison between the proposed protocol and that in [1] is shown in

Table 5. Although the number of encryption and multiplication in the proposed protocol is reduced compared to the protocol in [1], the proposed protocol needs $m\log p$ paring calculation. The computation cost of paring calculation is approximately 4 times than that of group calculation in the worst case [8]. Therefore, for the evaluation of efficiency, the proposed protocol is certainly more efficient than that in [1]. As for the communication costs, communication during Bidding and Opening phase in [1] and proposed protocol is the same, so it depends on the encrypted message sizes(that is, proportional to the key sizes) of each protocol.

A secure auction protocol for the first and second price auction was shown in [13]. However, in case of second price auction ($M = 1$), the proposed protocol is approximately twice faster than the one in [13]. In order to obtain the second highest bidding price, the protocol in [12] executes the first price auction protocol again after eliminating the highest bid.

5. Conclusion

We introduced new efficient secure $M + 1st$ 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 + 1-st price auction using homomorphic encryption," Proc. Public Key Cryptography, Lecture Notes in Computer Science, vol.2274, pp.115–124, 2002.
[2] D. Boneh and M. Franklin, "Efficient Generation of Shared RSA keys," Public Key Cryptography 1998, LNCS vol.1431, pp.1–13, 1998.
[3] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," Proc. Theory of Cryptography, Lecture Notes in Computer Science, vol.3378, pp.325–341, 2005.
[4] D.L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," Commun. ACM, vol.24, no.2, pp.84–90, 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, 1996.
[6] M. Jakobsson and A. Juels, "Mix and match: Secure function evaluation via ciphertexts," Advances in Cryptology, ASIACRYPT 2000, Lecture Notes in Computer Science, vol.1976, pp.162–177, 2000.
[7] A. Juels and M. Szydlo, "A two-server, sealed-bid auction protocol," Proc. Financial Cryptography, Lecture Notes in Computer Science, vol.2357, pp.72–86, 2003.
[8] T. Kerins, W.P. Marnane, E.M. Popovici, and P.S.L.M. Barreto, "Hardware accelerators for pairing based cryptosystems," IEE Proc. Inf. Secur., vol.152, no.1, pp.47–56, 2005.
[9] K. Kurosawa and W. Ogata, "Bit-slice auction circuit," Proc. Computer Security, ESORICS 2002, Lecture Notes in Computer Science, vol.2502, pp.24–38, 2002.
[10] H. Lipmaa, N. Asokan, and V. Niemi, "Secure vickrey auctions without threshold trust," Proc. Financial Cryptography, Lecture Notes in Computer Science, vol.2357, pp.87–101, 2003.
[11] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," Proc. 1st ACM Conference on Electronic Commerce, EC'99, pp.129–139, 1999.
[12] T. Nishide, M. Iwamoto, A. Iwasaki, and K. Ohta, "Secure (M + 1) st-price auction with automatic tie-break," Trusted Systems, Lecture Notes in Computer Science, vol.9473, pp.422–437, 2015.
[13] T. Mitsunaga, Y. Manabe, and T. Okamoto, "Efficient secure auction protocols based on the Boneh-Goh-Nissim encryption," IEICE Trans. Fundamentals, vol.E96-A, no.1, pp.68–75, Jan. 2013.
[14] T. Mistunaga, Y. Manabe, and T. Okamoto, "A secure M + 1st price auction protocol based on bit slice circuits," Proc. Advances in Information and Computer Security, Lecture Notes in Computer Science, vol.7038, pp.51–64, 2011.
[15] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," Advances in Cryptology, EUROCRYPT'98, Lecture Notes in Computer Science, vol.1403, pp.308–318, 1998.
[16] P. Pallier, "Public-key cryptosystems based on composite degree residuosity classes," Proc. Advances in Cryptology, EUROCRYPT'99, Lecture Notes in Computer Science, vol.1592, pp.223–238, 1999.
[17] C. Park, K. Itoh, and K. Kurosawa, "All/nothing election scheme and anonymous channel," Proc. Eurocrypt 1993, LNCS vol.765, pp.248–259, 1993.
[18] B. Rosenberg, ed., Handbook of Financial Cryptography and Security, Chapman and Hall/CRC, London, 2010.
[19] A. Yao, "Protocols for secure computations (exteded abstract)," Proc. FOCS'82, pp.160–164, 1982.

Takuho Mitsunaga received the B.Ec. and M.E. degrees from Osaka University and Kyoto University in 2008 and 2010, respectively. Currently, he is a project associate professor of Tokyo University. His research interests are cryptograpy and cyber security.

Yoshifumi Manabe received the B.E., M.E., and Dr.E. degrees from Osaka University, Osaka, Japan, in 1983, 1985, and 1993, respectively. From 1985 to 2013, he worked for Nippon Telegraph and Telephone Corporation. Since 2013, he is a professor at Kogakuin University. His research interests include distributed algorithms, cryptography, and graph theory. He was a guest associate professor of Kyoto University in 2001–2013. 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 Information Sharing Platform Laboratories. He is presently engaged in research on cryptography and information security. Dr. Okamoto is a guest professor of Kyoto University.