Regular Paper

Oblivious Comparator and Its Application to Secure Auction Protocol

Hiroaki Kikuchi†

This paper presents a protocol for Secure Function Evaluation (SFE) in which n players have secret inputs $E[a_1], E[a_2], \ldots, E[a_n]$, of a known boolean function f, and they collaborate to compute the ciphertext of the output of the boolean function, $E[f(a_1, \ldots, a_n)]$. The main result is a completeness theorem (Theorem 3.1) which states that an arbitrary function can be evaluated at the oblivious party without help of private information. The proposed protocol is based on the Jakobsson and Juels's Mix-and-Match scheme (Jakobsson and Jules, 2000) in which the truth table of a target function is row-wise randomized (*mixing*) using a Mix network, and then players perform "*matching*" the designated output ciphertext and the corresponding rows. The biggest difference between the proposed SFE and the Mix-and-Match is that the proposed protocol does not require any involvement of key holders to evaluate function, while the Mix-and-Match needs key holders to perform threshold decryptions at every step of evaluation of boolean gates. One disadvantage of the proposed scheme is the Reed-Muller expansion (Sasao, 1997) involves an exponential blow-up in the number of input, n, as the same as the conventional schemes, e.g., *CyptoComputer* proposed by Sander, Young, and Yung (1999). This paper presents an efficient construction for a primitive called '*oblivious comparator*' with n-round complexity between the comparator and n players but the bandwidth spent by one communication is independent from n (linear to the size of values to be compared), and hence it does not suffer the blow-up in n. The oblivious comparator is suitable to implement a secure auction because an auctioneer communicates with bidders once at time, and performs evaluation without help of trusted key holders. In addition, the proposed construction allows arbitrary complicated functions including a search for second highest, a resolution the winner, and a dynamic programming (for combinatorial auction).

1. Introduction

1.1 Secure Auction

Auctions in the electronic commerce are very complicated. The simplest auction style is the open-bid English auction, in which bidders incrementally raise the prices bid for goods until as many winners are left as the number of units of goods. As an alternative to this classical style of auction, an automatic agent system called "proxy bidding" [19] is becoming popular. A Dutch-style auction naturally satisfies the property that privacy of losing bids is preserved after auction closes. The Vickery auction, in which the winner who has the highest bid pays the second highest bid.

In the theory of economics, it is known that a social surplus is maximized when a bidder whose evaluation is highest wins the auction game and pay the uniform winning price which is independent of their evaluation. Wurman et al. proved that the $(M+1)$st-price auction satisfies a useful property, *incentive compatibility*, i.e., the dominant strategy is for a bidder to bid to his/her true valuation [20]. Since a winner's payment will be determined by the $(M+1)$st highest bid, which is the highest of all losing bids, every bidder who agrees to bid the maximum price he/she is willing to pay for a given item maximizes his/her chance to win without being worried that he/she might bid too much.

In order to satisfy the complicated requirements in secure auction, several attempts have been done in using cryptographical techniques. Franklin and Reiter present a sealed-bid auction protocol in Ref. 21). The protocol uses a verifiable signature sharing in order to prevent malicious bidder from canceling their bids. Bids are kept secret until the opening phase, and then all bids are opened and compared to determine the highest one.

Kikuchi, Hakavy and Tygar [22] improve the privacy of bids among distributed auctioneers even after the opening phase comes using a secure function computation of summation. The protocol runs in linear time to the number of possible bidding prices and cannot deal with tie breaking. In Ref. 24), Sako implements a Dutch-style auction using a distributed decryption. In the protocol, a bidder casts his bid encrypted by the public key corresponding to his bidding price. The privacy of losers' bits

† Department of Information Media Technology, School of Information Technology and Electronics, Tokai University

are kept under the assumption of not all auctioneers being faulty. Similarly, Miyazaki and Sakurai use an undeniable signature [25], and Kobayashi and Morita use an one-way hash chain [26].

Auctions in the electronic commerce are more complicated. Multiple buyers and sellers are involved and multiple unit of goods are auctioned in several environments. Wurman, Walsh and Wellman examined a several auction designs and analyzed in terms of the incentive compatibility in Ref. 20). They showed that the $(M+1)$st-price sealed-bid auction is incentive compatible for single-unit buyers. The secure second-price ($M=1$) auction protocol is presented by Hakavy, et al. [23]. They use the secure multiparty protocol of multiplication, presented in Ref. 13) in order to resolve the second highest bid in $O(k)$ rounds, where bids are represented as k-bit integers. Other papers on this subject include Refs. 27)∼32).

1.2 Secure Function Evaluation

The best way to securely implement arbitrary auction style is a Secure Function Evaluation (SFE). A SFE is a protocol to allow Alice and Bob, having inputs a and b, respectively, to compute a known function $f(a,b)$ while keeping their inputs private.

Goldreich, Micali, and Wigderson [11] presented a generalized scheme with an assumption of one-way trapdoor permutation, now commonly known as a secure multi-party computation (MPC). In their scheme, a target function is decomposed into gates represented as randomly permuted truth tables. Using a 1-2 oblivious transfer protocol, the other party picks the row of the truth table designated by his secret input bit. Although a target function is generally and systematically implemented using conventional circuit design technologies, intensive communication takes place to evaluate the function. The MPC scheme presented by Ref. 15), which is one of the most efficient MPCs, requires $O(n|C|)$ bits for broadcasting and $O(d)$ rounds for evaluation, where n is the number of parties, d is a depth of C and $|C|$ is the size of circuit, i.e., the number of gates in C.

Following the work by Ref. 11), a number of MPC schemes have been proposed. Ben-Or, Goldwasser and Wigderson presented an information-theoretically MPC scheme, in which an arbitrary function is realized as a composition of two arithmetical operations, addition and multiplication [13].

With the help of a cryptographic primitive known as verifiable secret sharing (VSS) [5], robustness against active adversaries is assured. One advantage of their approach compared with that of Ref. 11) is efficient arithmetic field operations rather than bitwise manipulations. A drawback is more assumptions, such as VSS, necessary to achieve robustness. Recently, Cramer, Damgård and Maurer showed that any linear secret sharing scheme allows general construction of an MPC scheme [14].

The Mix-and-Match scheme proposed by Jakobsson and Juels [3] is a new approach to SFE. In their scheme, rather than sharing private inputs into some players, ciphertext manipulation is used to evaluate the function. A brief description is as follows. A player provides ciphertexts of his private input bit. A target function f is decomposed into Boolean gates that are represented by a truth table, in which entries are publicly provided ciphertexts.

There are two steps, mixing and matching. In *mixing*, a row-wise permutation of the truth table is computed via a conventional mix network (e.g., Ref. 2)), and then players perform *matching*, in which some trusted parties look up the corresponding row in the mixed truth table and finally, without decrypting, the designated output ciphertext is obtained.

The interesting feature of the Mix-and-Match protocol is that only the key generation is distributed among trusted parties and an input does not have to be shared. The players do not even necessarily perform VSS and are not concerned about the other players. This feature makes performing the protocol simple and effective. Moreover, the target function can be easily constructed by boolean formulas as in the Ref. 11) approach.

A disadvantage of the Mix-and-Match protocol is a cryptographic primitive called a plaintext equality test ($\mathcal{PET}$), which allows players to determine whether two given ciphertexts represent the same plaintext. To perform the $\mathcal{PET}$, the all member of trusted authorities, having piece of distributed decryption key, have to jointly participate in the threshold decryption process. Moreover, the $\mathcal{PET}$ is required for each evaluation of bit in a boolean function to be evaluated and thus it spends bandwidth so much.

1.3 Our Contribution

In this paper, we propose a new SFE scheme based on the Mix-and-Match ciphertext manip-

ulation approach, but more suitable for practical secure auction. Rather than evaluating with the help of key holders, our protocol has an oblivious computer evaluate a target function without any knowledge of the private information. Therefore, every step of the computation can be made publicly verifiable; i.e., just showing all inputs and outputs, any party can ensure that the computer performs the correct manipulation. To prevent a malicious parity from misbehaving, some proof of knowledge protocols are used.

The main contribution of our paper is to prove that an arbitrary Boolean function can be evaluated without decrypting the input ciphertext within n rounds of iteration with players only encrypting once each. The well-known Boolean function canonical form called the Reed-Muller Expansion [1] is introduced. The advantage of this new protocol includes

- Non-interactivity. Players send their inputs to a server in a non-interactive fashion, i.e., they need not participate subsequently in evaluation of the function. This is significant requirement in auction.
- Oblivious Party. A server is an oblivious party without any secret information and hence is free from online attack, while some previous protocols such as Ref. 3) require intensive communication with parties owning secret information.

However, the protocol has a drawback of

- An exponential blow-up in message complexity. The Reed-Muller expansion requires all combination of inputs and thus the message size increases exponentially as the number of players n.

Nevertheless of the issue of the blowup, we believe the protocol is quite competitive with other secure function evaluation protocols because there is an appropriate application of the protocol in which the communication complexity is independent from n (linear to the size of values to be compared), and hence it does not suffer the blow-up in n. We present an efficient construction for a primitive called '*oblivious comparator*' with n-round complexity between the comparator and n players.

The oblivious comparator is suitable to implement a secure auction because an auctioneer communicates with bidders once at time, and performs evaluation without help of trusted key holders. In addition, the proposed construction allows arbitrary complicated functions including a search for second highest, a resolution the winner, and a dynamic programming (for combinatorial auction).

The outline of this paper is as follows. We generally describe a model and some building blocks in Section 2. In Section 3, we give the detail of our proposed protocol. As a practical application of our protocol, in Section 4, we present the *oblivious comparator*, which given n ciphertexts representing k-bit numbers generates the output ciphertexts in which the highest number and the identity of whose number has the highest number are encrypted. After computation at oblivious comparator, with the help of private key holders, the outputs are finally decrypted. The oblivious comparator cannot cheat bidders because of the proof of knowledge that she correctly performs comparison of bids but learn any knowledge from the result of the computation, namely, she does not know who is the winner nor how high the winning price is.

2. Model and Building Blocks

2.1 Model

We consider n players, $P_1, \ldots, P_n$, and some of these may be malicious. Each player a_i has a secret input a_i. A *computer* $\mathcal{C}$ takes n input ciphertexts, $E[a_1], \ldots, E[a_n]$, and outputs a ciphertext $E[y]$ of an agreed n-variable boolean function $y = f(a_1, \ldots, a_n)$. We assume a secure channel with confidentiality, sender authentication and message integrity between all pairs of the parties. We does not assume an authenticated broadcast channel in this model.

A computer $\mathcal{C}$ has a state S, which is a set of ciphertexts, and a state transition algorithm T, which takes an input ciphertext sent from a player and updates the state S according to the agreed function f. Every time $\mathcal{C}$ communicates with a player, the state S_i is updated to S_{i+1} and then referred by the subsequent player who uses the state to generate the next input ciphertext. Players communicate with $\mathcal{C}$ once, therefore, n rounds are involved to produce the final state S_n. After exactly n rounds between $\mathcal{C}$ and $P_1, \ldots, P_n$, $\mathcal{C}$ publishes the output ciphertext Y defined by a decoding algorithm D.

2.2 $\oplus$-Homomorphic Encryption Scheme

Let M be a set of plaintext, $\{m_0, m_1\}$, where m_0 and m_1 mean boolean values corresponding to 'false' and 'true', respectively.

Let us suppose a *homomorphic encryption scheme* E which satisfies the following proper-

ties:

- E is $\oplus$-homomorphic over $GF(2)$, i.e., for elements a and b of $\{m_0, m_1\}$,
$$E[a \oplus b] = E[a] \times E[b] \quad (1)$$
holds, where $a \oplus b$ is an Exclusive OR, defined as $m_0 \oplus m_1 = m_1 \oplus m_0 = m_1$ and $m_0 \oplus m_0 = m_1 \oplus m_1 = m_0$.
- E is semantically secure, that is, no one is able to distinguish ciphertexts of m_0 and m_1 with probability significantly greater than a random guess.
- The key generation can be distributed among a certain number of players. The size of the public key should not depend on the number of shares.
- The decryption process can be distributed among t-out-of-n players who share the corresponding private key. The computational and communicational (bandwidth and rounds) costs should be as small as possible.

The El Gamal encryption satisfies the all requirements under the Decision Diffie-Hellman (DDH) assumption, if we let $m_0 = 1$ and $m_1 = -1 \pmod p$. Let p and q be large primes such that $p = 2q + 1$ and $\mathcal{G}$ be the set of multiplicative groups of order q in Z_p^*. Let g be a primitive element of $\mathcal{G}$.

An El Gamal encryption of message m with public key $y = g^x$ is of the form $E_a[m] = (M, G) = (my^a, g^a)$, where a is a random number chosen from Z_q. To decrypt the ciphertext (M, G), we use the corresponding private key x to compute $M/G^x = mg^{xa-ax} = m$. By element-wise multiplication, we define $E[a] \times E[b] = (M_a M_b, G_a G_b)$, which yields a new ciphertext $E[a \oplus b]$, and it can be seen that the El Gamal encryption is $\oplus$-homomorphic over $GF(2)$.

Distributed decryption is also feasible. A private key is jointly generated by the collaboration of t honest parties (key holders) out of n and distributed among them using $(t-1)$-degree random polynomials $f(x)$ as $f(1)$, $f(2), \ldots, f(n)$. To decrypt a ciphertext provided with the public key $y = g^{f(0)}$, the i-th party publishes $G^{f(i)}$ for $i = 1, \ldots, t$, and then computes $G^{f(1)\gamma_1} \cdots G^{f(t)\gamma_t} = G^{f(0)}$ where γ_i is the LaGrange coefficient for i. For verifiability the players use Verifiable Secret Sharing (VSS) as proof of possession of $f(i)$ such that $G^{f(i)}$. See Ref. 5) for details.

An other instance of homomorphic encryption scheme is QR encryption as used in Refs. 8), 9).

2.3 Proof of Knowledge

We will use a proof of knowledge of private input to the computer, which is based on the disjunctive and conjunctive proofs of knowledge in Ref. 6).

Conjunctive Proof of Knowledge

By $PK\{(\alpha) : y_1 = g_1^\alpha \wedge y_2 = g_2^\alpha\}$, we denote a proof of knowledge of discrete logarithms of elements y_1 and y_2 to the bases g_1 and g_2. Selecting random numbers $r \in Z_q$, a prover sends $t_1 = g_1^r$ and $t_2 = g_2^r$ to a verifier, who then sends back a random challenge $c \in \{0,1\}^k$. The prover shows $s = r - c\alpha \pmod q$, which should satisfy both $g_1^s y_1^c = t_1$ and $g_2^s y_2^c = t_2$.

Disjunctive Proof of Knowledge

We denote by $PK\{(\alpha, \beta) : y_1 = g^\alpha \vee y_2 = g^\beta\}$ to mean a proof of knowledge of one out of the two discrete logarithms of y_1 and y_2 to the base g. Namely, the prover can prove that he knows a secret value under which either $y = y_1$ or $y = y_2$ must hold without revealing which identity was used. Without loss of generality, we assume that the prover knows α for which $y = g^\alpha$ holds. The prover uniformly picks $r_1, s_2 \in Z_q$ and $c_2 \in \{0,1\}^k$ and sends $t_1 = g^{r_1}$ and $t_2 = g^{s_2} y_2^{c_2}$ to the verifier, who then gives a random challenge $c \in \{0,1\}^k$, where k is a security parameter. On receiving the challenge, the prover sends $s_1 = r_1 - c_1\alpha \pmod q$, s_2, c_1 and c_2, where $c = c_1 \oplus c_2$. The verifier can see if the prover is likely to have the knowledge by testing both $t_1 = g^{s_1} y_1^{c_1}$ and $t_2 = g^{s_2} y_2^{c_2}$ with provability $1 - 2^{-k}$. Note that the same test can be used when t_1 and t_2 are prepared for the other knowledge β.

2.4 Reed-Muller Expansion

Let us review an ordinary version of Reed-Muller expansion defined with boolean values before we go to the cryptographic expansion.

Lemma 2.1 Let x, y and z be boolean values. Then, we have

1. $x \oplus y = y \oplus x$,
2. $0 \oplus x = x, 1 \oplus x = \overline{x}$,
3. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$,
4. $a(x \oplus y) = ax \oplus ay$,
5. $x \vee y = x \oplus y \oplus xy$.

Lemma 2.2 (Shannon Expansion) [1] Let $f(x_1, \ldots, x_n)$ be an n-variable boolean function. Then, f is expanded by

$$\begin{aligned} f &= \overline{x_1} F_0 \vee x F_1 \\ &= \overline{x_1} F_0 \oplus x F_1 \\ &= F_0 \oplus x_1 (F_0 \oplus F_1) \end{aligned} \quad (2)$$

where $F_0 = f(0, x_2, \ldots, x_n)$ and $F_1 = f(1,$

$x_2, \ldots, x_n)$.

By recursively applying Eq. (2) for every variable in f, we have the following canonical form of f.

Lemma 2.3 (Reed-Muller Expression)[1)] An arbitrary n-variable function $f(x_1, \ldots, x_n)$ is represented as

$$\begin{aligned} f = {} & a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n \\ & \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus \cdots \oplus a_{n-1\,n} x_{n-1} x_n \\ & \vdots \\ & \oplus a_{12\cdots n} x_1 x_2 \cdots x_n. \end{aligned} \tag{3}$$

Given a function, the boolean coefficients a_1, $a_2, \ldots, a_{12\cdots n}$ are uniquely determined. Equation (3) is called the Reed-Muller expression of f.

3. The Scheme

3.1 Logical Operations

We show that fundamental logical operations, conjunction ($\wedge$), disjunction ($\vee$) and negation ($\overline{x}$), are possible for input ciphertexts without decrypting.

Lemma 3.1 (negation) Let $E[a]$ be a ciphertext of a homomorphic encryption scheme and let a be an unknown element of $\{m_0, m_1\}$. Then, the ciphertext for negation of a is given by

$$E[\overline{a}] = E[a] \times E[m_1].$$

Proof. The proof is straightforward using Lemma 2.1. □

Lemma 3.2 (conjunction and disjunction) Let $E[a]$ be a ciphertext of a homomorphic encryption scheme where a is an unknown plaintext in $\{m_0, m_1\}$. Let us assume b is a known (private) plaintext in $\{m_0, m_1\}$. Then, the ciphertext of conjunction (AND) of a and b is obtained without learning a as follows:

$$E[ab] = \begin{cases} E[m_0] & \text{if } b = m_0, \\ E[a] & \text{if } b = m_1, \end{cases} \tag{4}$$

and the disjunction (OR) is

$$E[a \vee b] = \begin{cases} E[m_1] & \text{if } b = m_1, \\ E[a] & \text{if } b = m_0. \end{cases} \tag{5}$$

Proof. When $b = m_0$, the ciphertext $E[ab] = E[m_0]$ regardless of a. When $b = m_1$, the ciphertext $E[ab]$ is of m_1 only if $a = m_1$. The proof of disjunction is shown similarly. □

When the output ciphertext is required to be indistinguishable to anyone, including the owner of a, $E[a]$ in Eq. (4) can be re-encrypted using a random ciphertext $E[1]$ as $E[a]' = E[a] \times E[m_0]$.

Note that the plaintext a is not necessary to obtain $E[ab]$ and therefore conjunction consisting of multiple literals is feasible in the same manner. For example, from $E[a_1 a_2 a_3]$ and plaintext b, we can have $E[a_1 a_2 a_3 b]$.

Lemma 3.3 Let $E[a]$, $E[b]$ and $E[ab]$ be ciphertexts of secret plaintext in M. Then, the ciphertext of disjunction (OR) of a and b is

$$E[a \vee b] = E[a] \times E[b] \times E[ab].$$

Proof. The proof is shown immediately from Lemma 2.1. □

3.2 Basic Protocol

We consider an oblivious party $\mathcal{C}$ which has a set of ciphertexts of internal state and updates the state in a publicly verifiable manner. Before we consider a specific function that requires less communication cost, we more generally show functional completeness of ciphertext computation at an oblivious party $\mathcal{C}$.

Let S_i be a set of ciphertext which contains internal private state in $\mathcal{C}$. The i-th state set S_i is of the form $\{m_1, s_1, s_2, \ldots, s_{L_i}\}$, defined by $s_1 = E[a_1], s_2 = E[a_2], \ldots, s_{L_i} = E[a_1 a_2 \cdots a_i]$, and $L_i = |S_i| = 2^i$, where a plaintext a_i is either m_0 or m_1. In particular, let S_0 be $\{m_1\}$.

Basic Protocol (SFE)

1. Computer $\mathcal{C}$ sends to player P_i a current status $S_{i-1} = \{m_1, s_1, s_2, \ldots, s_{L_{i-1}}\}$.
2. For every element in S_{i-1}, player P_i use Eq. (4) in the conjunction protocol to compute the conjunctions of his private input a_i, and sends back to $\mathcal{C}$ the result $A_i = \{E[a_i], E[a_1 a_i], E[a_2 a_i], \ldots, E[a_1 a_2 \cdots a_{i-1} a_i]$. (Note that the constant m_1 in S_0 yields $E[m_1 a_i] = E[a_i]$ in A_i for every i.)
3. Computer $\mathcal{C}$ updates its state by $S_i = T(S_{i-1}, A_i)$.
4. Repeat 1 through 3 n times.
5. Computer $\mathcal{C}$ outputs $Y = D(S_n)$.

The transition algorithm T and the decoding algorithm D depend on the given boolean function f.

Theorem 3.1 (Functional Completeness) For an arbitrary n-variable boolean function $f(x_1, \ldots, x_n)$, there exists a transition algorithm T and a decoding algorithm D such that

$$\begin{aligned} & Y = D(S_n) = E[f(a_1, \ldots, a_n)], \\ & S_n = T(S_{n-1}, A_n) = T(T(S_{n-2}, A_{n-1}), A_n) \\ & \quad = \cdots = T(\cdots T(\{m_1\}, A_1) \cdots). \end{aligned}$$

Proof. By letting $T(S_{i-1}, A_i) = S_{i-1} \cup A_i$, we have the final state in which every element of power set of $\{a_1, a_2, \ldots, a_n\}$ is encrypted. Namely,

$S_n = \{m_1, E[a_1], E[a_2], \ldots, E[a_1 a_2 \cdots a_n]\}$.
From Lemma 2.3 and the homomorphism of encryption scheme, now we see that $\mathcal{C}$ has an arbitrary decoding algorithm D that produces the ciphertext of any given boolean function f. □

As an example of the decoding algorithm, let us consider a three-party majority function, which takes private boolean values a, b and c, and outputs whether more than two of them are 'true', or not. The basic protocol begins with $\mathcal{C}$ sending empty set to the first player who has a and sends back $A_1 = \{E[m_1 a]\} = \{E[a]\}$, which forms

$S_1 = S_0 \cup A_1 = \{m_1, E[a]\}$.

The second player computes the conjunction of $E[a]$ and her private b and sends to $\mathcal{C}$

$A_2 = \{E[b], E[ab]\}$,

which leads to $S_2 = S_1 \cup A_2$. Similarly, the interaction with the third player provides the final state

$$S_3 = \left\{ \begin{array}{l} E[a], E[b], E[c], \\ E[ab], E[ac], E[bc], E[abc] \end{array} \right\}.$$

Finally, the computer $\mathcal{C}$ invokes the Reed-Muller representation for the objective majority function as follows

$$\begin{aligned} D(S_3) &= E[ab] \times E[ac] \times E[bc] \times E[abc] \\ &= E[ab \vee ac \vee bc]. \end{aligned}$$

Note that $\mathcal{C}$ learns nothing from the outcome of his decoding algorithm under the assumption of an indistinguishable encryption scheme, e.g., DDH.

In the above description, the basic protocol is the simplest in the sense that players have just one bit secret, which can be easily extended to k-bit secret in a natural way. In some target boolean function, the transition algorithm can also be replaced by one requiring less storage. In a later section, we will show a variation of the basic protocol in which a player has a k-bit secret and a more lightweight transition algorithm is used.

3.3 Performance

Table 1 shows the message and rounds complexities of the proposed protocol in comparison with some previously proposed protocols for secure function evaluation. A message complexity is the number of bits sent or broadcasted in the evaluation. The notation of $|C|$ is the size of circuit C, i.e., the number of gates, and d is the depth of C.

In the perspective of message complexity, CDN01 [15] is the most efficient. However, it is not adequate to implement a certain application such as a sealed-bid auction because in the CDN01 all bidders are forced to participate in the processes to determine the winning price. This is not realistic as many bidders join the auction. On the other hand, the Mix-and-Match protocol achieves the same message complexity with allowing bidders to be off-line after submitting their bids, i.e., the *non-interactivity* is satisfied. It requires $O(n)$ rounds for submitting bids in a broadcast plus $O(d)$ rounds for computing C.

Table 1 Complexities of the proposed protocol.

protocol	message	rounds	circuit
BGW87 [13]	$\Omega(n^2\|C\|)$	$O(d)$	arithmetic
CDN01 [15]	$O(n\|C\|)$	$O(d)$	arithmetic
Mix and Match [3]	$O(n\|C\|)$	$O(n+d)$	boolean
Proposed	$O(2^n)$	$O(n)$	boolean

The proposed protocol also satisfies the non-interactivity as the Mix-and-Match protocol in the exponential expense of message complexity. The round complexity of $O(n)$ is not efficient but okay because we have seen that an ordinary single web server is able to handle many browsers one by one. In addition, the timeliness is not paramount requirement in many applications such as the sealed-bid auction. While, the exponential behavior of message complexity is critical. However, in the proceeding section, we show the application of the proposed protocol in which the message complexity is not exponential to n.

3.4 Verification Protocols

A malicious player may send an invalid input ciphertext to cheat other players or disrupt computation. To prevent players from violating the protocol without being detected, we require players to provide a proof of knowledge along with their ciphertexts. Although we have discussed a general homomorphic encryption scheme, we assume El Gamal encryption in this section.

Let $E[m] = (my^r, g^r) = (M, G)$ be a ciphertext encrypted with public key $y = g^x$. To prove $E[m]$ is valid ciphertext and m is either m_0 or m_1, a player who encrypts m shows a proof of knowledge of the form

$$PK\left\{(\alpha) : \begin{pmatrix} M = m_0 y^\alpha \\ \wedge G = g^\alpha \end{pmatrix} \vee \begin{pmatrix} M = m_1 y^\alpha \\ \wedge G = g^\alpha \end{pmatrix}\right\},$$

that is constructed based on the conjunctive and disjunctive proofs of knowledge in Ref. 6).

The proof of knowledge is not sufficient because a malicious player can send a valid ciphertext as $E[ab]$ which is inconsistent with

Table 2 Complexities of proof of knowledge.

ciphertext to be verified	message	computation proving	computation verifying	iteration
$E[a_i]$	$4\|p\|+2\|q\|$	6	8	1
$E[a_i s_j]$	$8\|p\|+4\|q\|$	12	16	L_i
total	$\|p\|(4+8L_i)+\|q\|(2+4L_i)$	$6+12L_i$	$8+16L_i$	

$E[a]$ and $E[b]$, e.g., claiming a forged ciphertext $E[m_0]$ to be $E[ab]$ but $a = m_1$ and $b = m_1$. To prevent players from casting $E[ab] = (M_{ab}, G_{ab})$ inconsistent with $E[a] = (M_a, G_a), E[b] = (M_b, G_b)$ and $E[m_0] = (M_0, G_0)$, we can force them to send a proof of knowledge $PK =$

$$\left\{ (\alpha,\beta): \begin{array}{l} \left(\begin{array}{l} M_a = m_0 y^\alpha \wedge \\ G_a = g^\alpha \wedge \\ M_{ab} = M_0 y^\beta \wedge \\ G_{ab} = g^\beta \end{array}\right) \\ \vee \left(\begin{array}{l} M_a = m_1 y^\alpha \wedge \\ G_a = g^\alpha \wedge \\ M_{ab} = M_b y^\beta \wedge \\ G_{ab} = g^\beta \end{array}\right) \end{array} \right\}.$$

Theorem 3.2 Under an assumption of computational Zero-knowledge proof, any dishonest player $(\mathcal{C}, P_1, \ldots, P_n)$ can not manipulate the result of computation in the basic protocol.

Let us estimate the overhead in terms of communication and computation given from the PK. Since the modular exponentiation is the dominant factor in proof of knowledge, omitting the other arithmetic such as multiplication we define a computation complexities as a number of modular exponentiations. For two types of proofs mentioned above, we show the result of estimation of message and computation complexities in **Table 2**, where $|p|$ is a size of p in bits and L_i is a number of element in i-th state set S_i defined in the basic protocol.

4. Application to Auction Protocol

4.1 Oblivious Comparator

In this section, we first consider a standard (non-cryptographical) version of a k-bit integer comparator and then extend it to an oblivious comparator that compares input ciphertexts without decrypting.

Given k-bit integers a and b such that $a = a_{k-1}2^{k-1} + \cdots + a_1 2^1 + a_0 2^0$ and $b = b_{k-1}2^{k-1} + \cdots + b_1 2^1 + b_0 2^0$, an oblivious comparator $\mathcal{C}$ wishes to have c such that $c = a$ if $a > b$; otherwise $c = b$. Obviously, c is a k-bit integer represented as $c = c_{k-1}2^{k-1} + \cdots + c_1 2^1 + c_0 2^0$. The design of the comparator is simple. For each two input bits a_i and b_i, the comparison is performed using three boolean variables, α_i, β_i and γ_i, such that α_i is true only if $a_i > b_i$, β_i is true only if $a_i < b_i$, and γ_i is true only if $a_i \neq b_i$. For example, we show the logic formulas of the comparator when $k = 3$ as follows:

$$\begin{aligned} c_2 &= a_2 \vee b_2, \\ \alpha_2 &= a_2 \overline{b_2}, \\ \beta_2 &= \overline{a_2} b_2, \\ \gamma_2 &= \alpha_2 \vee \beta_2, \\ c_1 &= \gamma_2(\alpha_2 a_1 \vee \beta_2 b_1) \vee \overline{\gamma_1}(a_1 \vee b_1), \\ \alpha_1 &= \alpha_2 \vee a_1 \overline{b_1}, \\ \beta_1 &= \beta_2 \vee \overline{a_1} b_1, \\ \gamma_1 &= \gamma_2 \vee \alpha_1 \vee \beta_1, \\ c_0 &= \gamma_1(\alpha_1 a_0 \vee \beta_1 b_0) \vee \overline{\gamma_1}(a_0 \vee b_0), \\ \alpha_0 &= \alpha_1 \vee a_0 \overline{b_0}, \\ \beta_0 &= \beta_1 \vee \overline{a_0} b_0, \\ \gamma_0 &= \gamma_1 \vee \alpha_0 \vee \beta_0. \end{aligned}$$

After the evaluation of the above logic formulas, the greater integer in a and b is given by (c_2, c_1, c_0) and the boolean variables α_0, β_0 and γ_0 indicate whether $a > b$, $a < b$, and $a \neq b$, respectively. Note that γ_0 is false if and only if $a = b$.

As we have seen in the basic protocol, an arbitrary boolean formula can be represented in the Reed-Muller expression. For example, the boolean variable $\gamma_2 = \alpha_2 \vee \beta_2 = a_2 \overline{b_2} \vee \overline{a_2} b_2$ has the Reed-Muller expression $a_2 \oplus b_2$. One more complicated example of c_1 is provided by using identities in Lemma 2.1 as follows:

$$\begin{aligned} c_1 &= \gamma_2(\alpha_2 a_1 \vee \beta_2 b_1) \vee \overline{\gamma_2}(a_1 \vee b_1) \\ &= \gamma_2(\alpha_2 a_1 \vee \beta_2 b_1) \oplus \overline{\gamma_2}(a_1 \vee b_1) \\ &= (a_2 \oplus b_2)(\alpha_2 a_1 \vee \beta_2 b_1) \\ &\quad \oplus (1 \oplus a_2 \oplus b_2)(a_1 \vee b_1) \\ &= a_1 \oplus b_1 \oplus a_1 a_2 \oplus a_1 b_1 \oplus a_2 b_1 \\ &\quad \oplus a_1 a_2 b_2 \oplus a_1 b_1 b_2 \oplus a_1 a_2 b_1 \oplus a_2 b_1 b_2 \\ &\quad \oplus a_1 a_2 b_1 b_2. \end{aligned}$$

Now, let us suppose that a player with a_2, a_1 and a_0 performs the three rounds of the basic protocol at once, i.e., sends to $\mathcal{C}$

$E[a_0], E[a_1], E[a_2], E[a_0 a_1], E[a_0 a_2], E[a_1 a_2], E[a_0 a_1 a_2]$.

The other player having b_0, b_1, b_2 also partic-

ipates in the protocol and provides a batch of ciphertexts

$$E[b_0], E[b_1], E[b_2], E[b_0a_0], E[b_0a_1], \ldots, E[b_0b_1b_2a_0a_1a_2].$$

As the result, $\mathcal{C}$ has many ciphertexts, sufficient to compose c_2, c_1, c_0 without decrypting.

The result of the comparison is still ciphertext, which can be used as an input ciphertext for subsequent comparison. Hence, the state transition algorithm for a k-bit comparator requires a constant size of internal state, which is independent of the number of integers to be examined. Indeed, we have

$$S_n = S_{n-1} = \cdots = S_1 = \{E[m_1], E[c_0], \ldots, E[c_0 \cdots c_k]\}.$$

The formal description of oblivious comparator is as follows.

Oblivious Comparator

1. A comparator $\mathcal{C}$ has an initial state $S_0 = \{s_0, \ldots, s_{2^k}\}$ such that $s_0 = s_1 = \cdots = s_{2^k} = E[m_1]$ and boolean variables $\alpha_{k+1} = \beta_{k+1} = \gamma_{k+1} = E[m_0]$.
2. Given state S_{i-1}, the i-th player with private value represented in $a_0, a_1, \ldots, a_k$ submits $A_i = \{$
$$E[m_1], E[a_0], \ldots, E[a_0 \ldots a_k],$$
$$E[s_1], E[a_0s_1], \ldots, E[a_0 \ldots a_ks_1],$$
$$\vdots$$
$$E[s_{2^k}], E[a_0s_{2^k}], \ldots, E[a_0 \ldots a_ks_{2^k}]$$
$\}$ in conjunction with the PK.
3. Comparator $\mathcal{C}$ updates the state by $S_i = T(S_i, A_i) = (s'_0, s'_1, \ldots, s'_{2^k}) = (m_1, c'_1, c'_2, \ldots, c'_k, c'_1c'_2, \ldots, c'_1c'_2 \cdots c'_k)$ such that
$$c'_j = \gamma_{j+1}(\alpha_{j+1}a_j \vee \beta_{j+1}c_j) \vee \overline{\gamma_{j+1}}(a_j \vee c_j),$$
$$\alpha_j = \alpha_{j+1} \vee a_j\overline{c_j},$$
$$\beta_j = \beta_{j+1} \vee \overline{a_j}c_j,$$
$$\gamma_j = \gamma_{j+1} \vee \alpha_j \vee \beta_j$$
for $j = k, k-1, \ldots, 0$ in the way of the basic protocol.
4. Comparator $\mathcal{C}$ repeats until $\mathcal{C}$ communicates with every player and outputs $Y = D(S_n) = (s_1, \ldots, s_k)$, which the trusted authorities decrypt and declare the highest value. Note that the decoding algorithm takes the first k element of S_n.

4.2 Sealed-Bid Auction Protocol

A suitable application of the oblivious comparator is a sealed-bid auction, where n bidders having k-bit private bids try to determine the highest bid and the winner in a secure manner. The oblivious comparator allows us to provide a trustworthy auctioneer who has interaction with every bidder once and blindly compare bids. The auctioneer has no chance to manipulate the winning price because all the processing steps, including a transition algorithm T and a decoding algorithm D, are publicly verifiable in the sense that anyone can make sure of the validity of the internal state without any secret information. The secrecy of the winning price and the winner are assured until more than a threshold number of trusted authorities who share the corresponding secret key agree to decrypt the resulting ciphertexts. With the non-interactive proof of sub-computations, the distributed decryption step is publicly verifiable in a secure manner, e.g., Ref. 18), and hence any dishonest behavior of authority can be detected.

In addition to functions for oblivious comparison, we need one more computer to determine the winner. This is not difficult. We suppose that every player has an assigned identity, say ID, that is a k_2-bit integer such that $k_2 > \log n$. The identities are encrypted as well and then sent to $\mathcal{C}$ in conjunction with input ciphertext. Using the boolean variables α_0 and β_0 in the oblivious comparator protocol, the comparator, $\mathcal{C}$, updates an additional internal state $W = (w_1, w_2, \ldots, w_{k_2})$ as

$$w_i = \alpha_0 IDA_i \vee \beta_0 IDB_i,$$

for $i = 1, \ldots, k_2$, where IDA_i and IDB_i are the i-th digits of identities for player A (bidder) and player B (internal state). When multiple bidders are tied at the same highest price, W is never set and therefore the default value will appear in the decrypted result as the sign that a tie occurred.

Using the above mentioned sub-protocols, we show a secure auction protocol as follows.

Secure Sealed-bid Auction (First-Price rule)

1. Auctioneer sets up a k-bit oblivious counter with initial state S_0 in which messages m_1 are encrypted by the a single or group of trusted authorities.
2. For each bidder $i = 1, \ldots, n$, the auctioneer performs the comparison protocol in Section 4.1 and updates the internal state S_i.
3. Bidders prove that their submitted bids $A_1, \ldots, A_n$ are correctly computed in the verification protocol in Section 3.4 and verify that the auctioneer updates the states correctly.
4. The auctioneer publishes the final state

Table 3 Complexities of the proposed secure sealed-bid auction protocols.

protocols	bidders			N.I.	servers		
	message		round				
	bids	PK			#	message	round
KHT98 [22]	$O(2^k)$	–	1	Y	m	$O(2^k)$	1
S00 [24]	$O(1)$	$O(1)$	1	Y	m	$O(n)$	$O(mk)$
MS99 [25]	$O(1)$	–	$O(k)$	N	1	$O(nk)$	$O(k)$
JJ00 [3]	$O(k)$	$O(k)$	1	Y	m	$O(knm)$	$O(d+n)$
Proposed	$O(2^k)$	$O(2^k)$	1	Y	1	$O(2^k n + km)$	$O(n)$

N.I. stands for Non-interactivity.

$Y = D(S_n)$ which contains the winning price and the ciphertext of the winner.

5. The trusted authorities jointly decrypt the ciphertexts to declare the winning price and the winner.

In the case of a second-highest-price auction, we construct the protocol by replacing the state transition algorithm T with that of keeping the second highest value. With the extension, the size of state is doubled.

4.3 Performance

In the proposed protocol, the amount of bit that a bidder has to send is the sum of 2^k ciphertexts for encoding bid and 2^{k_2} $(> \log n)$ ciphertexts for encoding identity, resulting the total of $(2^k + 2^{k_2})l$ where l is a size of ciphertext. By letting $l = 1024 \cdot 2$, $k = 10$ (that is, $2^k = 1024$ values are possible to assign as biding prices), we have the number of ciphertext for one bidder is 2048 and the expected time for sending is 0.2 second (in bandwidth of 1 Mbps).

If we take account of the cost for proof of knowledge, the expense increases. According to the performance analysis in Table 2, we estimate the message complexities of PK by assigning uniformly $L_1 = L_2 = \cdots = 2^k$ in k-bit comparison, as

$$|p|(4k + 8 \cdot 2^k) + |q|(2k + 4 \cdot 2^k) = O(2^k).$$

For instance of $k = 10$, $|p| = 1024$, a message size sent by a bidder is $1024 \cdot (40 + 8 \cdot 1024) \approx 10^6$ byte.

In **Table 3**, we summarize the performance of the proposed auction protocol in comparison with previously proposed auction protocols. The proposed protocol has an oblivious server and m trusted authorities who have distributed private key. The use of the proposed protocol is limited within small k.

5. Conclusion

We have proposed a protocol for Secure Function Evaluation (SFE) with ciphertext, in which n players with input ciphertexts collaborate to compute an output ciphertext of a known boolean function. The main result is that an arbitrary function evaluation is feasible without decrypting the input ciphertext in n rounds of communication with an oblivious computer. We have shown the oblivious comparator is suitable to construct a secure sealed-bid auction that satisfies privacy of bids, verifiability of bidders, accountability of auctioneers and efficient computation.

References

1) Sasao, T.: Easily Testable Realizations for Generalized Reed-Muller Expressions, *IEEE Trans. Computers*, Vol.46, No.6, pp.709–716 (1997).
2) Abe, M.: Universally Verifiable Mix-Net with Verification Work Independent of the Number of Mix-Servers, *IEICE Trans. Fundamentals*, Vol.E83-A, No.7 (July 2000).
3) Jakobsson, M. and Juels, A.: Mix and Match: Secure Function Evaluation via Ciphertexts, *Proc. ASIACRYPTO 2000*, LNCS 1967, pp.162–177 (2000).
4) Sander, T., Young, A. and Yung, M.: Non-Interactive CyptoComputing For NC1, *40th IEEE Annual Symposium on Foundations of Computer Science*, pp.554–567 (1999).
5) Pedersen, T.P.: A threshold cryptosystem without a trusted party, *Proc. EUROCRYPTO '91*, pp.522–526 (1991).
6) Cramer, R., Damgård, I. and Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols, *Proc. CRYPTO '94*, pp.174–187 (1994).
7) Camenisch, J. and Michels, M.: Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes, *Proc. EUROCRYPT'99*, pp.107–122 (1999).
8) Katz, J., Myers, S. and Ostrovsky, R.: Cryptographic Counters and Applications to Electronic Voting, *Proc.EUROCRYPT 2001*, LNCS 2045, pp.78–92 (2001).
9) Goldwasser, S. and Micali, S.: Probabilistic

Encryption, *Journal of Computer and System Sciences*, Vol.28, No.2, pp.270–299 (1984).
10) Cramer, R., Gennaro, R. and Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme, Proc. *EUROCRYPT 1997* (2001).
11) Goldwasser, S., Micali, S. and Wigderson, A.: How to Play Any Mental Game, or a Completeness Theorem for Protocols with an Honest Majority, *Proc. the Nineteenth Annual ACM STOC'87*, pp.218–229 (1987).
12) Beaver, D., Michali, S. and Rogaway, P.: The round complexity of secure protocols, *STOC*, pp.503–513 (1990).
13) Ben-Or, M., Goldwasser, S. and Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation, *STOC88*, pp.1–10 (1988).
14) Cramer, R., Damgård, I. and Maurer, U.: General Secure Multi-party Computation from any Linear Secret-Sharing Scheme, *Proc. EUROCRYPT 2000*, LNCS 1807, pp.316–334 (2000).
15) Cramer, R., Damgård, I. and Nielsen, J.B.: Multiparty Computation from Threshold Homomorphic Encryption, *Proc. EUROCRYPT 2001*, LNCS 2045, pp.280–300 (2001).
16) Beaver, D.: Minimal-Latency Secure Function Evaluation, *Proc. EUROCRYPT 2000*, LNCS 1807, pp.335–350 (2000).
17) Crescenzo, G. Di: Private Selective Payment Protocols, *Proc. Financial Cryptography 2000*, pp.72–89 (2000).
18) Gennaro, R., Jarecki, S., Krawczyk, H. and Rabin, T.: Robust threshold DSS signatures, *Proc. RUROCRYPT'96*, pp.354–371, Springer-Verlag, LNCS 1070 (1996).
19) eBay, `http://www.ebay.com`.
20) Wurman, P.R., Walsh, W.E. and Wellman, M.P.: Flexible Double Auctions for Electronic Commerce: Theory and Implementation, *Decision Support Systems*, Vol.24, pp.17–27 (1998).
21) Franklin, M.K. and Reiter, M.K.: The design and implementation of a secure auction service, *IEEE Trans. Softw. Eng.*, Vol.22, No.5, pp.302–312 (1996).
22) Kikuchi, H., Harkavy, M. and Tygar, J.D.: Multi-round anonymous auction, *IEICE Trans. Inf. & Syst.*, Vol.E82-D, No.4, pp.769–777 (1999).
23) Harkavy, M., Tygar, J.D. and Kikuchi, H.: Electronic auction with private bids, *Third USENIX Workshop on Electronic Commerce Proceedings*, pp.61–74 (1998).
24) Sako, K.: An auction protocol which hides bids of losers, *Proc. of PKC'2000*, pp.422–432 (2000).
25) Miyazaki, S. and Sakurai, K.: A bulletin board-based auction system with protecting the bidder's strategy, *Trans. IPSJ*, Vol.40, No.8, pp.3229–3336 (1999) (in Japanese).
26) Kobayashi, K. and Morita, H.: Efficient sealed-bid auction with quantitative competition using one-way functions, Technical Report of IEICE, *ISEC99-30*, pp.31–37 (1999).
27) Naor, M., Pinkas, B. and Sumner, R.: Privacy preserving auctions and mechanism design, *ACM Workshop on E-Commerce* (1999).
28) Stajano, F. and Anderson, R.: The cocaine auction protocol: on the power of anonymous broadcast, *Proc. Information Hiding Workshop 1999* (LNCS) (1999).
29) Cachin, C.: Efficient private bidding and auctions with an oblivious third party, *ACM Conference on Computer and Communications Security*, pp.120–127 (1999).
30) Stubblebine, S.G. and Syverson, P.F.: Fair On-Line Auctions without Special Trusted Parties, *Proc. Financial Cryptography 1999*, LNCS 1648, pp.230–240 (1999).
31) Kudo, M.: Secure electronic sealed-bid auction protocol with public key cryptography, *IEICE Trans. Fundamentals*, Vol.E81-A, No.1, pp.20–26 (1998).
32) Watanabe, Y. and Imai, H.: Optimistic Sealed-Bid Auction Protocol, *Proc. SCIS2000*, B09, pp.1–8 (2000).

(Received November 28, 2003)
(Accepted June 8, 2004)

Hiroaki Kikuchi was born in Japan. He received B.E., M.E. and Ph.D. degrees from Meiji University in 1988, 1990 and 1994. After he worked in Fujitsu Laboratories Ltd. from 1990 through 1993, he joined Tokai University in 1994. He is currently an Associate Professor in Department of Information Media Technology, School of Information Technology and Electronics, Tokai University. He was a visiting researcher of school of computer science, Carnegie Mellon University in 1997. His main research interests are fuzzy logic, cryptographical protocol, and network security. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE), the Information Processing Society of Japan (IPSJ), the Japan Society for Fuzzy Theory and Systems (SOFT), IEEE and ACM.