

Secure Generalized Vickrey Auction without Third-party Servers

Makoto Yokoo¹ and Koutarou Suzuki²

¹ NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
url: www.kecl.ntt.co.jp/csl/ccrg/members/yokoo/
e-mail: yokoo@cslab.kecl.ntt.co.jp

² NTT Information Sharing Platform Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka, Kanagawa 239-0847 Japan
e-mail: koutarou@isl.ntt.co.jp

Abstract. This paper presents a secure Generalized Vickrey Auction (GVA) scheme that does not require third-party servers, i.e., the scheme is executed only by an auctioneer and bidders. Combinatorial auctions, in which multiple goods are sold simultaneously, have recently attracted considerable attention. The GVA can handle combinatorial auctions and has good theoretical characteristics such as incentive compatibility and Pareto efficiency.

Secure GVA schemes have been developed to prevent frauds by an auctioneer. However, existing methods require third-party servers to execute the protocol. Having third-party servers that are operated by independent organizations is difficult in practice. Therefore, it is desirable that a protocol be executed by the participants themselves. However, if bidders take part in the execution of the auction procedure, a bidder might have an incentive to be an active adversary so that he manipulates the declarations of other bidders to become a winner or to decrease his payment. In our proposed scheme, we use a new protocol that can achieve the same outcome as the GVA. In this protocol, the procedure executed by a bidder affects neither the prices nor the allocation of the bidder. Therefore, a bidder does not have an incentive to be an active adversary. Also, the proposed scheme utilizes homomorphic encryption to prevent leaking private information among participants.

key words : generalized Vickrey auction, combinatorial auction, homomorphic encryption, mechanism design, game-theory.

category: research

1 Introduction

Combinatorial auctions have recently attracted considerable attention [15, 26, 29, 38, 39]. An extensive survey is presented in [8]. In contrast with conventional auctions that sell a single item at a time, combinatorial auctions sell multiple

items with interdependent values simultaneously and allow the bidders to bid on any combination of items.

In a combinatorial auction, a bidder can express complementary/substitutable preferences over multiple bids. For example, in the Federal Communications Commission (FCC) spectrum auction [19], a bidder could indicate his desire for licenses covering adjoining regions simultaneously (i.e., these licenses are complementary), while being indifferent as to which particular channel was awarded (channels are substitutable). By supporting such complementary/substitutable preferences, we can increase the bidder's utility and the revenue of the seller.

The Generalized Vickrey Auction (GVA) [34], which is also known as the Vickrey-Clarke-Groves (VCG) mechanism, is a generalized version of the well-known Vickrey auction [35] and one instance of the Clarke-Groves mechanism [7, 9]. The GVA can handle combinatorial auctions and has the following good theoretical characteristics.

Incentive Compatibility: For each bidder, truthfully declaring his evaluation values is the dominant strategy, i.e., the optimal strategy regardless of the actions of other bidders.

Pareto Efficiency: If all bidders take the dominant strategy (i.e., at the dominant strategy equilibrium), the social surplus, i.e., the sum of all participants' utilities including the auctioneer, is maximized.

Individual Rationality: No bidder suffers any loss by participating in the auction.

Also, under certain assumptions, we can show that only the GVA can satisfy all of these properties while maximizing the expected revenue of the auctioneer [17].

Although the GVA has these good theoretical characteristics, even its simplest form, i.e., the Vickrey auction, is not yet widely used. As discussed in [25], the main difficulty of using the Vickrey auction is its vulnerability to an insincere auctioneer. For example, if the highest bid is \$1,000 and the second highest bid is \$500, then, the payment of the winner becomes \$500. However, by fabricating a dummy bid at \$999, the auctioneer can increase his revenue to \$999.

Another difficulty is that the true evaluation value is sensitive information that a bidder may not want to reveal [25]. For example, if a company wins in a public tender, then its bidding value, i.e., its true cost, becomes public, and the company may have difficulty in negotiating with sub-contractors.

The authors have developed a secure GVA scheme [33] that utilizes homomorphic encryption. However, this scheme requires that each bidder declare his evaluation values for all m^n possible allocations, where m is the number of goods and n is the number of bidders. This is inevitable for implementing the GVA in the most general case. However, for many auctions in the real world, we can assume the following two conditions.

No allocative externality: Each bidder is only concerned with the goods that are allocated to him, and he is indifferent to the allocations of other bidders.

Free disposal: Goods can be discarded without any cost.

In this case, declaring his evaluation values for m^n possible allocations is grossly redundant, since his evaluation value is the same as long as the goods allocated to him are the same. Also, he needs to declare his evaluation values only for the bundles in which he is interested.

The authors have developed secure Dynamic Programming (DP) protocols [32, 40]. In principle, by repeatedly solving winner-determination problems, we can obtain the results of the GVA. However, the disadvantage of these secure DP protocols, as well as the scheme presented in [33], is that they require third-party servers. These servers must be operated by independent organizations to avoid collusion among the servers. In practice, collecting a large number of such servers is very difficult. Therefore, it is desirable that the protocol be executed without such third-party servers. Pier-to-pier network services that do not require central servers are becoming popular, since such services are more robust against various failures and are scalable. Similarly, for combinatorial auctions, it is desirable that the auction be executed only by participants, i.e., the auctioneer and bidders, without using third-party servers.

However, if bidders take part in the execution of the auction procedure, even if the auction protocol is incentive compatible, a bidder might have an incentive to be an active adversary. For example, in the GVA, a bidder can manipulate the evaluation values of other bidders so that he would be a winner or his payment would be decreased. We can avoid such manipulations by making the procedure publicly verifiable. However, this requires additional communication/processing costs.

In this paper, we develop a new auction protocol that can obtain the same outcome as the GVA. In this protocol, for each bidder, the price of each set of goods (bundle) is calculated first, and then each bidder can choose a bundle that maximizes his utility based on these prices independently from the choices of bundles of other bidders. This protocol looks quite different from a standard GVA description, in which an allocation is determined first, then the payment is calculated. However, we show that our new protocol can obtain exactly the same outcome as the GVA.

The advantage of this new protocol is that its procedures can be distributed among bidders without giving them incentives to be an active adversary. More specifically, in this protocol, the prices and allocation of bidder j is determined independently from the prices of bidder i . Therefore, if bidder j participates in the procedure for calculating the prices of bidder i , bidder j does not have an incentive to be an active adversary who manipulates the prices of bidder i .

For example, in the most simplest form of this protocol, in which a single unit of a single good is auctioned, the price of bidder i for the good is defined as the maximal evaluation value among all bidders other than i . Clearly, this protocol is identical to the Vickrey auction protocol. The task of calculating the price of bidder i can be distributed among other bidders, since even if bidder j manipulates bidder i 's price (so that the price is increased), this manipulation does not affect the price of bidder j . Therefore, bidder j does not have an incentive to be an active adversary who manipulates the price of bidder i .

We apply dynamic programming (DP) [3] to calculate the prices. Also, our proposed secure GVA scheme utilizes homomorphic encryption to prevent leaking private information, i.e., evaluation values, to other bidders.

The rest of this paper is organized as follows. First, we describe the standard description of the GVA in Section 2. Next, in Section 3, we describe our newly developed protocol that can obtain the same outcome as the GVA. Then, in Section 4, we describe the method for obtaining the prices required in this protocol using dynamic programming (DP) [3]. Next, we describe the details of the proposed secure GVA scheme in Section 5. Finally, we discuss the related works and remaining issues, including collusion among bidders, in Section 6.

2 Preliminaries: GVA

This section describes the standard description of the GVA. First, we define several terms and notations.

- $N = \{1, 2, \dots, n\}$: a set of bidders
- $M = \{1, 2, \dots, m\}$: a set of goods
- $u(i, B)$: the valuation of bidder i for bundle B
- We assume free-disposal, i.e., for all $B \subset B'$, $u(i, B) \leq u(i, B')$ holds.
- We assume quasi-linear utility, i.e., if bidder i obtains B by paying p_i , his utility is given by $u(i, B) - p_i$.
- $\mathbf{g} = (g(1), g(2), \dots, g(n))$: a feasible allocation \mathbf{g} for a set of goods M , where $g(i)$ represents the bundle allocated to bidder i , and the following two conditions hold: $\bigcup_i g(i) \subseteq M$, and for all $i \neq j$, $g(i) \cap g(j) = \emptyset$
- $G(M)$: a set of all feasible allocations of goods M .

In the GVA, for each bundle B , bidder i declares his evaluation value $v(i, B)$. Note that the declared evaluation value is not necessarily the same as the true evaluation value $u(i, B)$. The protocol selects a Pareto efficient allocation based on the declared evaluation values. More specifically, we choose an allocation $\mathbf{g}^* = (g^*(1), g^*(2), \dots, g^*(n)) \in G(M)$ so that for any allocation $\mathbf{g} = (g(1), g(2), \dots, g(n)) \in G(M)$, the following condition holds.

$$\sum_j v(j, g^*(j)) \geq \sum_j v(j, g(j)).$$

There might be multiple allocations that are Pareto efficient. In that case, the GVA arbitrary selects one Pareto efficient allocation \mathbf{g}^* .

Next, the payment of bidder i is determined as follows. Let us assume $\mathbf{g}_{\sim i}^* = (g_{\sim i}^*(1), g_{\sim i}^*(2), \dots, g_{\sim i}^*(n)) \in G(M)$ is an allocation defined as follows. For any allocation $\mathbf{g} = (g(1), g(2), \dots, g(n)) \in G(M)$, the following formula holds.

$$\sum_{j \neq i} v(j, g_{\sim i}^*(j)) \geq \sum_{j \neq i} v(j, g(j)).$$

The payment of bidder i , i.e., p_i , is defined as follows.

$$p_i = \sum_{j \neq i} v(j, g_{\sim i}^*(j)) - \sum_{j \neq i} v(j, g^*(j)).$$

In the GVA, for each bidder i , declaring the true evaluation values, i.e., declaring $v(i, \cdot) = u(i, \cdot)$, is a dominant strategy, i.e., the best strategy to maximize his utility regardless of the actions of other bidders. This property is called *incentive compatibility*. The reason that the GVA is incentive compatible is explained as follows. Since the utility of i is quasi-linear, it can be represented as follows.

$$u(i, g^*(i)) - p_i = u(i, g^*(i)) - \left[\sum_{j \neq i} v(j, g_{\sim i}^*(j)) - \sum_{j \neq i} v(j, g^*(j)) \right] \quad (1)$$

$$= [u(i, g^*(i)) + \sum_{j \neq i} v(j, g^*(j))] - \sum_{j \neq i} v(j, g_{\sim i}^*(j)) \quad (2)$$

The third term of formula (2) is determined independently from bidder i 's declarations. Therefore, bidder i can maximize his utility by maximizing the sum of the first and second terms of formula (2). On the other hand, g^* is selected by maximizing $\sum_j v(j, g^*(j)) = v(i, g^*(j)) + \sum_{j \neq i} v(j, g^*(j))$. This means bidder i can maximize his utility by declaring $v(i, \cdot) = u(i, \cdot)$, i.e., by declaring true evaluation values.

Let us describe how the GVA works. Assume there are two goods 1 and 2 and three bidders 1, 2, and 3. The evaluation value for a bundle $u(i, B)$ is given as follows.

	{1}	{2}	{1, 2}
bidder 1	6	0	6
bidder 2	0	0	8
bidder 3	0	5	5

In a Pareto efficient allocation, good 1 is allocated to bidder 1 and good 2 is allocated to bidder 3. The payment of bidder 1 is calculated as follows. Without considering bidder 1's evaluation value, the best allocation is to allocate both goods to bidder 2, and the sum of the evaluation values is 8. When considering bidder 1, the sum of the evaluation values other than that of bidder 1 is 5. Therefore, the payment of bidder 1, i.e., p_1 , is given as $8 - 5 = 3$. Similarly, the payment of bidder 3 is given as $8 - 6 = 2$.

3 New GVA-equivalent Protocol

In this section, we develop a new protocol that can achieve the same outcome as the GVA. In this protocol, as in the standard GVA, for each bundle B , bidder i

declares his evaluation value $v(i, B)$. Note that the declared evaluation value is not necessarily the same as the true evaluation value $u(i, B)$.

To simplify the protocol description, we introduce the following notation. For a set of goods $B \subseteq M$ and a set of bidders X , we define $V^*(B, X)$ as the sum of the evaluation values of X when B is allocated optimally among X . To be precise, let us represent the set of all feasible allocations of a set of goods B as $G(B)$, where for each $\mathbf{g} = (g(1), g(2), \dots, g(n)) \in G(B)$, $\bigcup_{i \in X} g(i) \subseteq B$ and for all $i \neq j$, $g(i) \cap g(j) = \emptyset$ holds. $V^*(B, X)$ is defined as follows.

$$V^*(B, X) = \max_{\mathbf{g} \in G(B)} \sum_{j \in X} v(j, g(j)).$$

In this protocol, instead of determining the allocation first, we first determine the price of each bundle B for each bidder i . The price of bundle B for bidder i is defined as follows.

$$p_{i,B} = V^*(M, N \setminus \{i\}) - V^*(M \setminus B, N \setminus \{i\}).$$

Next, each bidder i chooses a bundle that maximizes his utility based on the prices, i.e., he chooses B_i^* , where $B_i^* = \arg \max_{B \subseteq M} u(i, B) - p_{i,B}$. Note that each bidder can choose a bundle that maximizes his utility independently from the choices of other bidders. To be more precise, if there exist multiple bundles that maximize his utility, then the protocol performs some adjustment so that the choices are consistent, but each bidder is still guaranteed to obtain one bundle that maximizes his utility.

It is obvious that this new protocol satisfies incentive compatibility. For bidder i , his prices are determined independently from i 's declaration. Also, he can choose the optimal bundle regardless of the choices of other bidders. Therefore, bidder i has no incentive to manipulate the prices of other bidders (which are dependent on his declaration). Since this protocol satisfies incentive compatibility, in the rest of this paper, we assume each bidder declares his true evaluation values $u(i, B)$.

This protocol is identical to the GVA, i.e., the following theorems holds.

Theorem 1. *A bundle B maximizes bidder i 's utility if and only if for some \mathbf{g}^* , $g^*(i) = B$ holds.*

Theorem 2. *If B maximizes bidder i 's utility, then $p_i = p_{i,B}$ holds.*

In proving these theorems, we use the following characteristics. From the definition, the following formula holds.

$$\sum_{j \neq i} u(j, g_{\sim i}^*(j)) = V^*(M, N \setminus \{i\}).$$

Furthermore, for $\mathbf{g}^* = (g^*(1), g^*(2), \dots, g^*(n))$, the following formula holds.

$$\sum_{j \neq i} u(j, g^*(j)) = V^*(M \setminus g^*(i), N \setminus \{i\}).$$

The proof of Theorem 1 is as follows. First, we show if for some \mathbf{g}^* , $g^*(i) = B$, then B maximizes bidder i 's utility. More specifically, we are going to derive a contradiction by assuming for some \mathbf{g}^* , $g^*(i) = B$ but bundle B does not maximize bidder i 's utility. In this case, there exists another bundle B' and $u(i, B') - p_{i, B'} > u(i, B) - p_{i, B}$ holds.

$$p_{i, B'} = V^*(M, N \setminus \{i\}) - V^*(M \setminus B', N \setminus \{i\}).$$

$$p_{i, B} = V^*(M, N \setminus \{i\}) - V^*(M \setminus B, N \setminus \{i\}).$$

Therefore, the following formula holds.

$$u(i, B') + V^*(M \setminus B', N \setminus \{i\}) > u(i, B) + V^*(M \setminus B, N \setminus \{i\}).$$

However, the right side of this equation can be transformed as follows.

$$\begin{aligned} u(i, B) + V^*(M \setminus B, N \setminus \{i\}) &= u(i, g^*(i)) + V^*(M \setminus g^*(i), N \setminus \{i\}) \\ &= u(i, g^*(i)) + \sum_{j \neq i} u(j, g^*(j)) \\ &= \sum_j u(j, g^*(j)). \end{aligned}$$

The right side of this equation represents the sum of evaluation values at Pareto efficient allocation \mathbf{g}^* . On the other hand, the left side is the sum of evaluation values when allocating B' to bidder i and allocating other goods optimally among bidders other than i . This contradicts the assumption that \mathbf{g}^* is Pareto efficient.

Next, we prove that if a bundle B maximizes bidder i 's utility, then for some \mathbf{g}^* , $g^*(i) = B$ holds. More specifically, we are going to derive a contradiction by assuming a bundle B maximizes bidder i 's utility but for any \mathbf{g}^* , $g^*(i) \neq B$.

In this case, there exists bundle B' , where $B' \neq B$, $B' = g^*(i)$, and $u(i, B) - p_{i, B} > u(i, B') - p_{i, B'}$ hold. Therefore, the following formula holds.

$$u(i, B) + V^*(M \setminus B, N \setminus \{i\}) > u(i, B') + V^*(M \setminus B', N \setminus \{i\}).$$

However, the right side of this formula represents the sum of evaluation values at Pareto efficient allocation \mathbf{g}^* , while the left side is the sum of evaluation values when allocating B to bidder i and allocating other goods optimally among bidders except i . This contradicts the assumption that \mathbf{g}^* is Pareto efficient. \square .

Next, we prove Theorem 2. From Theorem 1, when B maximizes bidder i 's utility, then for some \mathbf{g}^* , $g^*(i) = B$ holds.

$$\begin{aligned} p_i &= \sum_{j \neq i} u(j, g_{\sim i}^*(j)) - \sum_{j \neq i} u(j, g^*(j)) \\ &= V^*(M, N \setminus \{i\}) - V^*(M \setminus g^*(i), N \setminus \{i\}) \\ &= p_{i, B}. \end{aligned}$$

If there exist multiple Pareto efficient allocations, then multiple bundles can simultaneously maximize the bidder's utility. In this case, the protocol needs to

adjust allocations so that the choices of bidders are consistent, i.e., no good is allocated to different bidders simultaneously. However, Theorem 1 states that any bundle B that is allocated to bidder i in a Pareto efficient allocation would maximize bidder i 's utility. Therefore, by choosing any Pareto efficient allocation, we can find a way to adjust choices so that the choices of bidders are consistent and each bidder is guaranteed to obtain one of the optimal bundles.

Let us describe how this protocol works. In the identical setting of the previous example, the price of each bundle is calculated as follows.

	{1}	{2}	{1, 2}
bidder 1	3	8	8
bidder 2	6	5	11
bidder 3	8	2	8

As a result, bidder 1 obtains good 1 at price 3, and bidder 3 obtains good 2 at price 2.

4 Calculating Prices using Dynamic Programming

In this section, we show a method for calculating the prices of bidder i by using dynamic programming (DP) [3]. This method is based on the method for solving winner determination problems in a combinatorial auction described in [26].

We assume each bidder j (except i) is declaring his evaluation value $u(j, B)$ for each bundle B in which he is interested. If bidder j has substitutable evaluation values, e.g., bidder j wants B_1 or B_2 but not both at the same time, we introduce a dummy good d . More specifically, we assume bidder j is interested in both $B_1 \cup \{d\}$ and $B_2 \cup \{d\}$. By introducing the dummy good, we can avoid allocating both B_1 and B_2 to bidder j at the same time.

Then, we create a node $(B, |B|)$ for each bundle $B \subseteq M$. $|B|$ is the number of goods included in B . Also, we create the following directed, weighted links for each bundle B in which bidder j is interested.

- a link from $(B, |B|)$ to $(\{\}, 0)$, where its weight $w((B, |B|), (\{\}, 0))$ is $u(j, B)$
- for each $B', B'' \subseteq M$, where $B'' \subset B'$, $B' \setminus B'' = B$, and $|B''| \geq |B'|/2$, a link from $(B', |B'|)$ to $(B'', |B''|)$, where its weight $w((B', |B'|), (B'', |B''|))$ is $u(j, B)$

If there exists bundle B in which nobody is interested, then we assume a dummy bidder j_d is interested in B , where evaluation value $u(j_d, B) = 0$. We show an example of nodes and links, where $M = \{1, 2, 3\}$, in Figure 1.

In this graph, the length of the longest path from node $(B, |B|)$ to terminal node $(\{\}, 0)$ represents the sum of the evaluation values when allocating goods B optimally to bidders other than i , i.e., $V^*(B, N \setminus \{i\})$.

Let us represent the length of the longest path from $(B, |B|)$ as $f((B, |B|))$. Then, $f((B, |B|))$ can be defined by the following recurrence formula.

- $f(\{\}, 0) = 0$
- $f((B, |B|)) = \max_{((B', |B'|))} w((B, |B|), (B', |B'|)) + f((B', |B'|))$.

By using this formula, we can obtain $f((B, |B|))$ by starting from a node that has smaller $|B|$.

The price of bidder i for bundle B , i.e., $p_{i,B}$, is given as $V^*(M, N \setminus \{i\}) - V^*(M \setminus B, N \setminus \{i\})$. Therefore, $p_{i,B} = f((M, |M|)) - f((M \setminus B, |M \setminus B|))$.

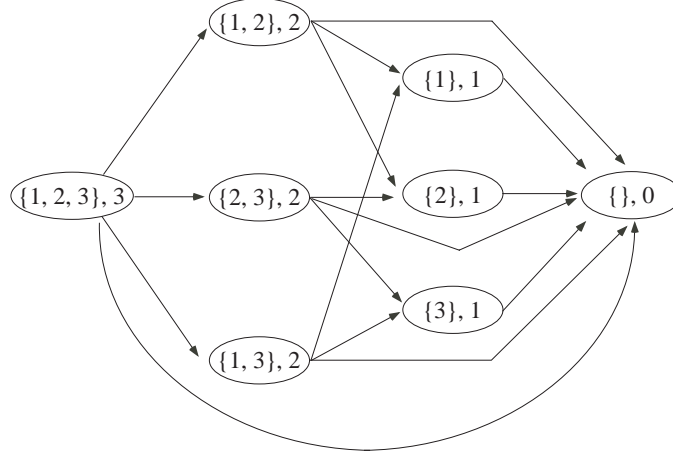


Fig. 1. Example of graph for Dynamic Programming

One important special case of general combinatorial auctions is multi-unit auctions, in which multiple identical units of a good is auctioned. In this case, as described in [32], the DP procedure requires only $O(n \times m)$ nodes instead of 2^m nodes.

5 Secure GVA

In this section, we show the details of the proposed secure GVA scheme. This scheme utilizes homomorphic encryption to prevent leaking private information among participants.

5.1 Preliminaries: Homomorphic Encryption and Vector Representation

Let E be a probabilistic public key encryption that provides indistinguishability, homomorphic property, and randomizability. The homomorphic property means that $E(a)E(b) = E(ab)$, and the randomizability means that one can compute a randomized ciphertext $E'(a)$ only from the original ciphertext $E(a)$, i.e., without

knowing either the decryption key or the plaintext. For instance, ElGamal encryption or Paillier encryption [23] have these desired properties, so our auction scheme can be built on these encryption schemes.

We represent the weight of a link in the following way so that we can determine the maximum of weights, and can add a constant to a weight. This method is based on the technique used in [40].

Vector representation: We represent a weight w ($1 \leq w \leq w_{max}$) by vector $\mathbf{e}(w)$ of ciphertexts

$$\mathbf{e}(w) = (e_1, \dots, e_{w_{max}}) = (\underbrace{E(z), \dots, E(z)}_w, \underbrace{E(1), \dots, E(1)}_{w_{max}-w}),$$

where $E(1)$ and $E(z)$ denote the encryption of 1 and common public element $z (\neq 1)$, respectively. Here, $\text{ord}(z)$ and w_{max} are chosen to be large enough. Because of the indistinguishability of E , we cannot determine w without decrypting each element.

Find the maximum: We can find the maximum of encrypted weights $\mathbf{e}(w_i) = (e_{1,i}, \dots, e_{w_{max},i})$ without leaking any information about the weights that are not the maximum as follows. Consider the componentwise product of all vectors

$$\prod_i \mathbf{e}(w_i) = (\prod_i e_{1,i}, \dots, \prod_i e_{w_{max},i}).$$

Observe that, due to the homomorphic property, the j -th component of this vector has the following form

$$c_j = \prod_i e_{j,i} = E(z^{S(j)}),$$

where $S(j) = \#\{i \mid j \leq w_i\}$ is the number of weights that are equal to or greater than j . Notice that $S(j)$ monotonically reduces as j increases. To find the maximum of these weights, we decrypt c_j and check whether decryption $D(c_j)$ is equal to 1 or not from $j = w_{max}$ to $j = 1$ until we find the largest j s.t. $D(c_j) \neq 1$. This j is equal to $\max_i \{w_i\}$, i.e., the maximum of the weights.

Add a constant: We can add a constant c to encrypted weight $\mathbf{e}(w) = (e_1, \dots, e_{w_{max}})$ without learning w . By shifting and randomizing $\mathbf{e}(w)$, we can obtain

$$\mathbf{e}'(w + c) = (\underbrace{E(z), \dots, E(z)}_c, e'_1, \dots, e'_{w_{max}-c}),$$

where e'_j is a randomization of ciphertext e_j . Due to randomization, one can obtain no information about constant c from $\mathbf{e}(w)$ and $\mathbf{e}'(w + c)$. Note that we can perform these operations without decrypting $\mathbf{e}(w)$ nor learning w .

Using these methods, we can find the maximum of weights, and add a constant to a weight; thus we can perform dynamic programming procedures using this vector representation.

5.2 Proposed Scheme

The outline of the proposed secure GVA scheme is as follows.

- Each bidder is given a share of a secret key for E so that if t bidders cooperate, they can decrypt E . In the preparation phase, the secret and public keys are generated in a distributed way [24], and each bidder has only a share of the secret key.
- Each bidder i posts a vector that represents evaluation value $u(i, B)$ for each bundle B in which he is interested to a common bulletin board.
- For bidders $i = 1, 2, \dots, n$, the following procedures are executed.
 1. t bidders other than i are assigned to calculate the prices for bidder i . We call these bidders as price-calculators for bidder i .
 2. Price-calculators construct the graph using the evaluation values except that of i . Then, they calculate $f((B, |B|))$ for each node using the method described in the previous subsection. Next, for each bundle B in which bidder i is interested, they calculate $p_{i,B} = f((M, |M|)) - f((M \setminus B, |M \setminus B|))$.
 3. Price-calculators send prices to bidder i .
 4. Bidder i finds all bundles that maximize his utility and posts the set of bundles to the bulletin board.
- The auctioneer adjusts bundle allocations so that they do not conflict with each other.

As described in Section 3, we can always find consistent bundle allocations.

5.3 Security

Privacy Loss: By using the proposed scheme, a price-calculator for bidder i learns $f((B, |B|))$, i.e., $V^*(B, N \setminus \{i\})$ for each node $(B, |B|)$. However, $f((B, |B|))$ is obtained by aggregating many evaluation values of bidders. Therefore, it is difficult to precisely estimate each evaluation value from $f((B, |B|))$. Unless t (or more) participants collude, they cannot decrypt vectors that represent the evaluation values of bidders directly.

Incentive to be Active Adversary: If bidder j is a price-calculator for bidder i , the prices and the allocation of bidder i do not affect the prices or the allocation of bidder j , i.e., bidder j can obtain the optimal bundle regardless the allocation of bidder i . Therefore, bidder j does not have an incentive to be an active adversary and to manipulate the prices of bidder i . Of course, a bidder might have an incentive to alter the evaluation values of other bidders posted on the bulletin board so that his prices are decreased. We need to utilize signature schemes to prevent such manipulations.

5.4 Efficiency

In the proposed scheme, each bidder declares his evaluation values only for the bundles in which he is interested. On the other hand, in the scheme described

in [33], each bidder needs to declare his preference over m^n possible allocations, where n is the number of bidders and m is the number of goods. When the number of bidders n becomes large, the number of possible allocations becomes very large.

In the proposed scheme, for each bidder i , we need to calculate the prices for all bundles in which i is interested using the DP procedure. The number of nodes is 2^m . When performing the DP procedure, price-calculators need to communicate with each other to decrypt the vector that represents $e(f((B, |B|)))$ for each node $(B, |B|)$. The number of decryptions for each node is at most w_{max} , i.e., the number of elements of each vector. Therefore, the total number of communications required for decryption in total is $n \cdot 2^m \cdot w_{max} \cdot t$.

6 Discussions

Many papers consider secure sealed-bid auctions. Kikuchi, Harkavy and Tygar presented an anonymous sealed-bid auction that uses encrypted vectors to represent bidding prices [13]. They also proposed a Vickrey auction, where the bidding price is represented by polynomials that are shared by auctioneers [11]. Kudo used a time server to realize sealed-bid auctions [18]. Cachin proposed a sealed-bid auction using homomorphic encryption and an oblivious third party [5]. Sakurai and Miyazaki proposed a sealed-bid auction in which a bid is represented by the bidder's undeniable signature of his bidding price [28]. Stubblebine and Syverson proposed an open-bid auction scheme that uses a hash chain technique [30]. Naor, Pinkas and Sumner realized a sealed-bid auction by combining Yao's secure computation with oblivious transfer [20]. Juels and Szydlo improved this scheme [12]. Sako proposed a sealed-bid auction in which a bid is represented by an encrypted message with a public key that corresponds to his bidding price [27]. Kobayashi, Morita and Suzuki proposed a sealed-bid auction that uses only hash chains [31, 16]. Omote and Miyaji proposed a sealed-bid auction that is efficient [22]. Watanabe and Imai proposed a sealed-bid auction that utilizes a chain of encryption [36]. Kikuchi proposed an $M + 1$ -st price auction, where the bidding price is represented by the degree of a polynomial shared by auctioneers [14]. Baudron and Stern proposed a sealed-bid auction based on circuit evaluation using homomorphic encryption [2]. Chida, Kobayashi and Morita proposed a sealed-bid auction with low round complexity [6]. Abe and Suzuki proposed an $M + 1$ -st price auction using homomorphic encryption [1]. Lipmaa, Asokan and Niemi proposed an $M + 1$ -st price auction without threshold trust [10]. Omote and Miyaji proposed an $M + 1$ -st price auction using the p -th residue problem [21]. Suzuki and Yokoo proposed a combinatorial auction that uses secure dynamic programming [32, 40]. Brandt proposed a $M + 1$ -st price auction where bidders compute the result by themselves [4].

In all of these schemes, however, the GVA has not been treated with remarkable exceptions [20] and [33]. Naor, Pinkas and Sumner [20] proposed a general method for executing any auction, including combinatorial auctions, based on a technique called the garbled circuit [37]. This method does not require interactive

communications among multiple evaluators. However, designing a combinatorial circuit to implement GVA is still an open problem, and the obtained circuit can be prohibitively large.

Suzuki and Yokoo [33] proposed a secure GVA protocol. However, as discussed in Section 1, this scheme requires that each bidder declare his evaluation values for all m^n possible allocations. Also, this scheme requires third-party servers.

By repeatedly applying the secure dynamic programming scheme [32, 40], we can determine the winner and payments of the GVA. However, if a bidder participates in the traditional procedure of the GVA, the bidder might have an incentive to be an active adversary.

When bidders take part in the auction procedure, an auctioneer might be concerned about collusion among bidders. Actually, the GVA protocol is vulnerable against collusions even if the auctioneer executes the auction procedure. For example, in the simplest form of the GVA, i.e., the Vickrey auction, bidder i , who has the highest evaluation value, can bribe the second highest bidder j to reduce j 's declaration so that i 's payment becomes smaller. One method to reduce the effect of collusion is to set reservation/minimal prices for each good. In our proposed scheme, this is possible if the auctioneer participates in the auction as a bidder. The auctioneer declares a price of each good, i.e., he is not willing to sell the good at less than the price.

In our proposed scheme, we need to create a graph that consists of 2^m nodes to calculate the prices. If the number of goods m becomes large but the number of bundles in which each bidder is interested is relatively small, the graph contains exponentially many nodes, while most of the links are dummy links with zero weights. We are currently developing an efficient method for handling such graphs. Also, as mentioned in Section 4, in multi-unit auctions, which are an important subclass of general combinatorial auctions, we need to create only $O(n \times m)$ nodes.

7 Conclusions

In this paper, we developed a secure GVA scheme in which the GVA can be executed without using third-party servers, i.e., the scheme can be executed only by the auctioneer and bidders. We first developed a new auction protocol that can achieve the same outcome as the GVA. In this protocol, for each bidder, the price of each bundle is calculated first, then each bidder can choose a bundle that maximizes his utility based on these prices independently from the choices of bundles of other bidders. In this protocol, the prices and allocation of bidder j are determined independently from the prices of bidder i . Therefore, if bidder j participates in the procedure for calculating the prices of bidder i , bidder j does not have an incentive to be an active adversary who manipulates the prices of bidder i . The proposed scheme utilizes the secure DP protocol to calculate the prices, which prevents leaking evaluation values of a bidder to other participants.

References

1. M. Abe and K. Suzuki. M+1-st price auction using homomorphic encryption. *Proceedings of Public Key Cryptography 2002*, 2002.
2. O. Baudron and J. Stern. Non-interactive private auctions. *Proceedings of Fifth International Financial Cryptography Conference (FC-01)*, 2001.
3. R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
4. F. Brandt. Fully private auctions in a constant number of rounds. *Proceedings of Seventh International Financial Cryptography Conference (FC-2003)*, 2003.
5. C. Cachin. Efficient private bidding and auctions with an oblivious third party. *Proceedings of 6th ACM Conference on Computer and Communications Security*, pages 120–127, 1999.
6. K. Chida, K. Kobayashi, and H. Morita. Efficient sealed-bid auctions for massive numbers of bidders with lump comparison. *Proceedings of ISC 2001*, 2001.
7. E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 2:19–33, 1971.
8. S. de Vries and R. V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15, 2003.
9. T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
10. N. A. H. Lipmaa and V. Niemi. A two-server, sealed-bid auction protocol. In *Proceedings of Sixth International Financial Cryptography Conference (FC-02)*, 2002.
11. M. Harkavy, J. D. Tygar, and H. Kikuchi. Electronic auctions with private bids. *Proceedings of Third USENIX Workshop on Electronic Commerce*, pages 61–74, 1998.
12. A. Juels and M. Szydlo. A two-server, sealed-bid auction protocol. In *Proceedings of Sixth International Financial Cryptography Conference (FC-02)*, 2002.
13. H. Kikuchi, M. Harkavy, and J. D. Tygar. Multi-round anonymous auction protocols. *Proceedings of first IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, pages 62–69, 1998.
14. H. Kikuchi. (M+1)-st-Price auction protocol. In *Proceedings of Fifth International Financial Cryptography Conference (FC-01)*, 2001.
15. P. Klemperer. Auction theory: A guide to the literature. *Journal of Economics Surveys*, 13(3):227–286, 1999.
16. K. Kobayashi, H. Morita, K. Suzuki, and M. Hakuta. Efficient sealed-bid auction by using one-way functions. *IEICE Trans. Fundamentals*, E84-A(1), 2001.
17. V. Krishna. *Auction Theory*. Academic Press, 2002.
18. M. Kudo. Secure electronic sealed-bid auction protocol with public key cryptography. *IEICE Trans. Fundamentals*, E81-A(1), 1998.
19. J. McMillan. Selling spectrum rights. *Journal of Economics Perspectives*, 8(3):145–162, 1994.
20. M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, pages 129–139, 1999.
21. K. Omote and A. Miyaji. A second-price sealed-bid auction with the discriminant of the p-th root. In *Proceedings of Sixth International Financial Cryptography Conference (FC-02)*, 2002.
22. K. Omote and A. Myaji. An anonymous auction protocol with a single non-trusted center using binary trees. *Proceedings of ISW2000*, pages 108–120, 2000. LNCS 1975.

23. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Proceedings of EUROCRYPT '99*, pages 223–238, 1999.
24. T. Pedersen. A threshold cryptosystem without a trusted party. *Proceedings of EUROCRYPT '91*, pages 522–526, 1991. Lecture Notes in Computer Science 547.
25. M. H. Rothkopf, T. J. Teisberg, and E. P. Kahn. Why are vickrey auctions are rare. *Journal of Political Economy*, 98(1):94–109, 1990.
26. M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
27. K. Sako. Universally verifiable auction protocol which hides losing bids. *Proceedings of Public Key Cryptography 2000*, pages 35–39, 2000.
28. K. Sakurai and S. Miyazaki. A bulletin-board based digital auction scheme with bidding down strategy. *Proceedings of 1999 International Workshop on Cryptographic Techniques and E-Commerce*, pages 180–187, 1999.
29. T. Sandholm. An algorithm for optimal winner determination in combinatorial auction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 542–547, 1999.
30. S. G. Stubblebine and P. F. Syverson. Fair on-line auctions without special trusted parties. *Proceedings of Third International Financial Cryptography Conference (FC-99)*, 1999.
31. K. Suzuki, K. Kobayashi, and H. Morita. Efficient sealed-bid auction using hash chain. *Proceedings of International Conference Information Security and Cryptology 2000 (LNCS 2015)*, pages 183–191, 2000.
32. K. Suzuki and M. Yokoo. Secure combinatorial auctions by dynamic programming with polynomial secret sharing. In *Proceedings of Sixth International Financial Cryptography Conference (FC-02)*, Lecture Notes in Computer Science 2357, pages 44–56. Springer, 2002.
33. K. Suzuki and M. Yokoo. Secure Generalized Vickrey Auction using homomorphic encryption. In *Proceedings of Seventh International Financial Cryptography Conference (FC-03)*, 2003.
34. H. R. Varian. Economic mechanism design for computerized agents. In *Proceedings of the First Usenix Workshop on Electronic Commerce*, 1995.
35. W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
36. Y. Watanabe and H. Imai. Reducing the round complexity of a sealed-bid auction protocol with an off-line ttp. *Proceedings of ACM Conference on Computer and Communications Security 2000*, pages 80–86, 2000.
37. A. C. Yao. How to generate and exchange secrets. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.
38. M. Yokoo, Y. Sakurai, and S. Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130(2):167–181, 2001.
39. M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior*, forthcoming.
40. M. Yokoo and K. Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 112–119, 2002.