


RESEARCH

Open Access



# Privacy-preserving combinatorial auction without an auctioneer

Chunqiang Hu<sup>1,2\*</sup> , Ruinian Li<sup>3</sup>, Bo Mei<sup>3</sup>, Wei Li<sup>4</sup>, Arwa Alrawais<sup>3</sup> and Rongfang Bie<sup>5</sup>

## Abstract

Combinatorial auctions are employed into many applications such as spectrum auctions held by the Federal Communications Commission (FCC). A crucial problem in such auctions is the lack of secure and efficiency mechanism to protect the privacy of the bidding prices and to ensure data security. To solve the problem, we propose an approach to represent the price as a polynomial's degree based on verifiable secret sharing. So, we can obtain the two polynomials's degree maximum/sum by the degree of the two polynomial's degree sum/product. In the protocol, the bidders' information is hidden. The auctioneers can receive the shares without a secure channel, so our protocol is more applicable to more scenarios. The scheme can resist the collusion attack, passive attack and so on. Moreover, Compared to Kikuchi (IEICE Trans Fundam Electron Commun Comput Sci 85(3):676–683, 2002); Suzuki and Yokoo (Secure combinatorial auctions by dynamic programming with polynomial secret sharing, 2003), the proposed scheme has the authentication property without increasing the communications cost.

**Keywords:** Security, Verification, Combinatorial auctions, Dynamical programming, Secret sharing

## 1 Introduction

Recently, combinatorial auctions have become an interesting domain, which allow that multiple goods are sold simultaneously and any combination of goods can be bid. For example, FCC spectrum, network routing, and railroad segment can be auctioned.

To carry out a combinatorial auction, the winner determination problem has to be solved first. The problem can be cooperatively solved by multi-auction servers, which can calculate the maximum sum of combinations of bidding prices. It is a challenge problem to protect bidding prices. If the auctioneer is trust, it can solve the winner determination problem. However, it is not practical as the auctioneer may collude with a participant to reveal the bids' information during the auction. If a strategy-proof mechanism is utilized to resist collusion attacks. However, the auctioneer can create a fake bid to increase revenue.

In traditional auctions, cryptographic functions (public key cryptography, hash chains, etc.) [1–4] are utilized to protect the bid's privacy. However, these schemes do not

consider spatial reuse, so they are not applicable to the secondary spectrum market. In the secondary spectrum market, SPRING was proposed in [5], which introduces a trustworthy agent to interact with both the auctioneer and the bidders. The sensitive information can be protected. However, SPRING depends on a trusted third party (the agent). In [6–10], homomorphic encryption [11–13] is employed to hid each bidder's bidding values with a vector of cipher texts, and ensures the auctioneer to figure out the maximum value, and charge the bidders securely. However, the homomorphic encryption has a higher computational cost, which is not practical now.

To tackle the above challenges, two problems have to be solved. First, multi-auction servers compute the maximum sum of combinations of bidding prices, while the information of bids and the part of the optimal solution should be kept secret. Second, the collusion activity of multi-auction servers must be resisted. We employ verifiable secret sharing [14] to protect privacy and data security in combinatorial auctions. The scheme allows multi-servers to randomly choose secret shares and verify the legitimacy of them to each other.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 presents preliminaries. In Section 4, we describe the main idea of the proposed

\*Correspondence: [hcq0394@gmail.com](mailto:hcq0394@gmail.com)

<sup>1</sup>Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing, China

<sup>2</sup>School of Software Engineering, Chongqing University, Chongqing, China  
Full list of author information is available at the end of the article

scheme. In Section 5, we analyze the security and performance of the scheme, followed by a conclusion in Section 6.

## 2 Related works

To protect data security and privacy in auctions, cryptographic tools, such as AES, homomorphic encryption, and secret sharing, have been applied.

SPRING [5] presents a trustworthy agent to protect the sensitive information of the auctions. However, SPRING depends on a trusted third party (the agent). In [6, 7], the authors utilize a vector of cipher text to mask the bidding prices, and guarantee that the maximum value, randomizing the bids, and charging the bidders can be figured out. However, the schemes [6, 7] are not practical as homomorphic encryption has a very high computational overhead, which is not applicable to the applications now. In [8], a secure auction without auctioneer scheme for VCG auction is designed based on homomorphic, in which the bidders work together to decide who the winner is without auctioneer; however, the computational overhead is high for each bidder, which has low efficiency. In [9], the authors design a sealed-bid first-price auction scheme based on homomorphic encryption, in which the server processes the bidder's encrypted bids using homomorphic encryption and the aggregation result is known by auctioneer; however, the scheme cannot resist the collusion activity between the server and auctioneer.

In [15, 16], the bidding prices are hidden via secret sharing. However, there are two weaknesses in [15] as follows. First, the relationships of multi-winner can not be solved. Second, the scheme is not efficient as the computational cost is very much higher. The bids are hidden by the degree of polynomials [16]. However, the scheme is based on the passive adversary model and cannot resist collusion attacks. Therefore, it is not practical.

In [17, 18], the sealed-bid auctions are constructed via verifiable secret sharing. The scheme can resist collusion attacks among the evaluators. However, the secret shares are obtained from a third party via a private secure channel, so the scheme cannot resist collusion attacks amongst evaluators and the third party.

In this paper, we present a privacy-preserving combinatorial auction without an auctioneer based on verifiable

secret sharing [14]. Compared to [15, 16], it does not need a secure channel among the bidders and the server. Meanwhile, the proposed scheme provides the authentication without increasing the communication cost.

## 3 Preliminaries

We now introduce some preliminary concepts for the cryptographic primitives used in this paper.

### 3.1 Dynamic programming

Dynamic programming [19] can be utilized to solve the problem, which is viewed as the result of a sequence of stepwise decisions.

We first describe the dynamic programming's concept via an algorithm of finding the longest path in a one-dimensional-directed graph in Fig. 1. The graph includes the nodes  $S, 1, 2, \dots, n$  with directed links among them. The link is denoted  $(j, k)$ , where  $j < k$ .  $w(j, k)$  denoted the weight for each link  $(j, k)$ . Figuring out the longest path from initial node  $S$  to terminal node  $n$  is our goal, i.e., to find a maximized path from  $S$  to  $n$ . For the sake of simplicity, we assume that it exists at least one link from  $j$  for each node  $j$  (where  $1 = j < n$ ) except node  $n$ .

We assume the longest path from  $S$  to  $n$  is denoted by  $L$ . The last half of  $L$  for any node  $j$  on  $L$  is also a longest path from  $j$  to  $n$ , which is called the principle of optimality. We can utilize the feature to search out the original problem's optimal solution via the sub-problems' optimal solutions.

Specifically, the longest path from  $S$  to  $n$  can be obtained by figuring out the following recurrence formula from node  $n - 1$  to  $S$ . In the formula, the longest path from  $j$  to  $n$  is denoted as  $f(j)$ .  $f(j)$  is called the node  $j$ 's evaluation value.  $f(n)$  is defined as  $S$  for terminal node  $n$ .  $f(S)$  represents the optimal solution for initial node  $S$ .

$$f(j) = \max_{(j,k)} \{w(j, k) + f(k)\} \tag{1}$$

When we calculate the formula, the value  $f(j)$  of the link  $(j, k)$  is recorded for each node  $j$ , i.e.,  $\max_{(j,k)} \{w(j, k) + f(k)\}$  is the value of the link, which recorded links from  $S$  to  $n$  constructs the longest path.

Assume that there are  $n + 1$  stages  $j = 1, \dots, n$  and each stage  $j$ 's state is  $(j, s)$ . When  $j < k$ , there can be directed links  $((j, s), (k, t))$  between these states. The

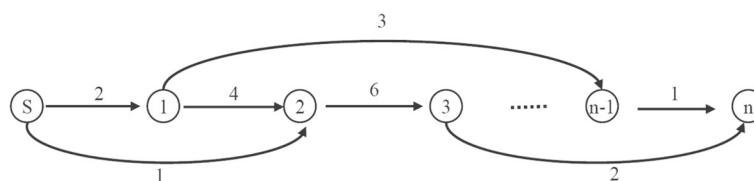


Fig. 1 An example of one dimension directed graph

weight  $w((j, s), (k, t))$  is given for each link. The following recurrence formula is defined dynamic programming evaluates function  $f$ :

$$f((j, s)) = \max_{j < k, ((j, s), (k, t))} \{w((j, s), (k, t)) + f((k, t))\} \quad (2)$$

The evaluation value  $f((S, s))$  can be calculated, which is the original problem's optimal value, by iteratively applying the relation for  $j = n, n - 1, \dots, 1$  with initial values  $f((n, s)) = iv(s)$ .

We introduce the proposed privacy-preserving combinatorial auction without an auctioneer based on the longest path of a one-dimensional directed graph. An example is introduced in Section 4.6.

### 3.2 Secret sharing schemes

Secret sharing is an important cryptographic primitive, which is utilized to our scheme. Since secret sharing is developed by Shamir [20] and Blakley [21] in 1979, many secret sharing schemes have been extensively studied [14, 22–24]. Generally speaking, secret sharing is briefly introduced as follows. A *dealer* shares a secret with a number of *users*  $U_1, \dots, U_n$ , a user gets the secret if and only if it can co-work with at least  $t - 1$  other users, where  $t \leq n$  is a pre-determined parameter. The dealer shares the secret and the users is  $s \in GF(p_1)$ , where  $p_1 > N$ . Each user  $U_i$  holds a secret key  $k_i \in GF(p_1)$ , which is only known by  $U_i$  and the dealer.

The dealer follows two step procedure. First, it constructs a polynomial function  $F(x)$  of degree  $t - 1$  shown in (3):

$$F(x) = s + \sum_{j=1}^{t-1} \mu_j x^j, \quad (3)$$

by randomly choosing each  $\mu_j$ . Note that all (additive and multiplication) operations used in (3) is modular arithmetic (defined over  $GF(p_1)$ ) as opposed to real arithmetic. Also  $s$  forms the constant component of  $F(x)$  - i.e.,  $s = F(0)$ . Then, in the second step, the dealer sends a shared secret  $s_i = F(x_i)$  to each  $U_i$ , where  $x_i$  is a random number selected by  $U_i$  and is sent to the dealer via the secure channel protected by  $k_i$ .

We now show how to recover  $s$  by  $t$  or more users. Without loss of generality, let  $U_1, \dots, U_t$  be the cooperating users. The secret  $s = F(0)$  can be reconstructed from  $s_1 = F(x_1), \dots, s_t = F(x_t)$  by these  $t$  users.

$$s = F(0) = \sum_{j=1}^t \left( s_j \prod_{i \in [1, t], i \neq j} \frac{0 - x_i}{x_j - x_i} \right). \quad (4)$$

Note that the cumulative product in (4) is essentially the Lagrange coefficient. The correctness of (4) can be easily verified based on the definition of  $F(x)$ .

## 4 The proposed scheme—secure computing

We present the proposed privacy-preserving combinatorial auction without an auctioneer, and we also discuss the security and efficiency of the scheme.

### 4.1 Requirements

The requirements for the secure protocol are as follows:

1. Evaluators (servers) select their secret keys by themselves, and the weight publishers (WP) (buyers and sellers) calculate and publish the weights for each share.
2. The legitimacy of evaluators is verified to each other, and then the evaluators cooperatively implement dynamic programming protocol to find the optimal solution, while each weight is kept secret.

To achieve this goal, the following two questions should be solved: How to resist collusion attacks? How to figure out the maximum sum of weights without revealing each weight? We denote a weight as a polynomial's degree; So, the degree of the sum/product of the two polynomials construct the maximum/sum of the degree of two polynomials, and verifiable secret sharing scheme [14, 25] is employed to resist collusion attack.

### 4.2 Basic idea

Weight publisher  $WP$  has a secret  $s \in Z_N$ .  $WP$  chooses random  $n$  ( $n > s$ ) points  $x_1, x_2, \dots, x_n \in Z_N$ , the constant  $c \in Z_N$ , and publish them. Then, it randomly chooses a polynomial  $A \in Z_N[x]$  s.t.  $deg(A) = s$  and  $A(0) = c$  and holds its secret.  $WP$  publishes its shares  $\{A(x_1), A(x_2), \dots, A(x_n)\}$ . Each evaluator  $E_l$  holds its share for  $A(x_l)$ , where  $l$  is the number of the evaluators,

A masking polynomial  $M \in Z_N[x]$  s.t.  $deg(M) = d$  and  $M(0) = 0$  is chosen by each  $WP$ , who keeps it secret. Then,  $WP$  calculates its  $l$  shares  $M(x_l)$ , and  $l - th$  share is selected by each evaluator. Then, masked shares  $A(x_l) + M(x_l)$  where ( $l = 1, 2, \dots, d + 1$ ) are published by  $d + 1$  evaluators  $\{E_1, E_2, \dots, E_{d+1}\}$ . The evaluators utilize these  $d + 1$  masked shares to perform polynomial interpolation, i.e., determine polynomial is  $A + M$ , recover  $A(0) = A(0) + M(0)$ , and verify whether  $A(0) = c$  or not. We can recover the constant term  $A(0) = c$  from  $d + 1$  shares if  $deg(A) = d$ , where  $deg(A + M) = d$ . We cannot recover the constant term  $A(0) = c$  from  $d + 1$  shares if  $deg(A + M) > d$ . Thus, we are convinced that  $deg(A) = d$  if  $A(0) = c$  holds. Furthermore, the degree of the sum/product of the two polynomials can construct using the maximum/sum of the degree of two polynomials by the following formulas:

$$\max\{deg(A), deg(B)\} = deg(A + B) \quad (5)$$

$$deg(A) + deg(B) = deg(A \cdot B) \quad (6)$$

The maximum/sum of two secrets to be locally determined as each evaluator  $E_l$  can calculate its share of sum  $A + B$  / product  $A \cdot B$  of two polynomials  $A$  and  $B$  by calculating the sum  $A(x_l) + B(x_l)$  / product  $A(x_l) \cdot B(x_l)$  of two shares  $A(x_l)$  and  $B(x_l)$ .

**4.3 System model**

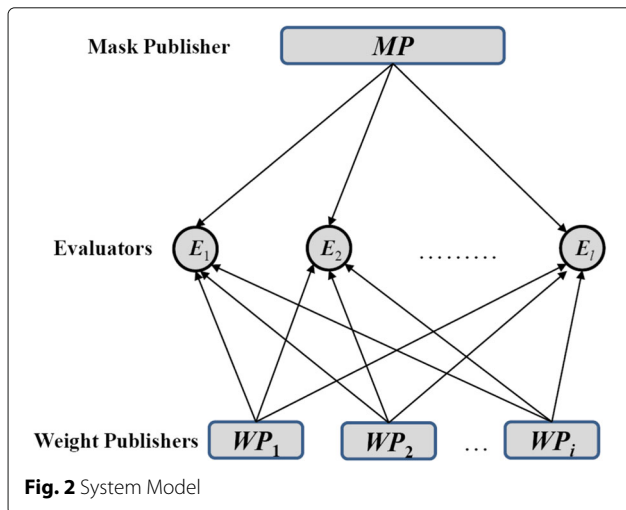
As shown in Fig. 2, our system model consists of three major entities: mask publisher (MP), evaluators (E), weight publishers (WP). In the following, we briefly summarize the major functions of each entity.

- *Mask publisher (MP):* MP is used to generate and distribute keys for all evaluators. MP also generates the mask polynomial, and distributes the mask value for each evaluator.
- *Evaluators (E):* Each evaluator cooperatively executes dynamic programming and finds the optimal solution and verifies the identities of evaluators each other.
- *Weight publishers (WP):* Each WP distributes its shares to each evaluators.

**4.4 Security model**

In our security model, we consider that the following security goals need to be achieved:

- *Privacy-preservation of bidders' bids.* The evaluators should be able to verify the identities of other evaluators; i.e, when the evaluators work together to figure out the optimal solution, they should verify the identities of each participant fist; meanwhile, the privacy should be protected.
- *Non-repudiation:* any bidder (weight publisher) cannot repudiate his bid.



**Fig. 2** System Model

- *Accountability:* any bidder can be verified that they follow the protocol to get the optimal solution by the evaluators.

**4.5 Secure computing**

**4.5.1 Initialization phase**

There is a mask publisher,  $MP$ , which chooses a randomly masked polynomial  $M \in Z_N[x]$  s.t.  $deg(M) = d$  and  $M(0) = 0$  and keeps it secret. The weight publishers  $WP_{(i,j)}$  for each link  $(i, j)$ . There are  $l$  evaluators  $\{E_1, E_2, \dots, E_l\}$  where  $l$  is greater than the length of the longest path.

To solve the verification problem, the inter-communication is needed by the mask publisher  $MP$  and the evaluators. The communication between  $MP$  and the evaluators can use the public channel. First, the mask publisher randomly selects two strong primes  $p$  and  $q$ , and calculates  $N = pq$ . Then, the mask publisher figures out the generator  $g$ , and publishes  $\{g, N\}$ .

Each evaluator  $E_i$  randomly chooses an integer  $s_i$  as its secret share where  $s_i \in [2, N]$ , and calculates  $R_i = g^{s_i} \text{ mod } N$ . Then,  $E_i$  sends  $R_i$  and its identity number  $id_i$  to mask publisher  $MP$ . For any two pair of evaluators  $E_i$  and  $E_j$ ,  $MP$  must guarantee that  $R_i \neq R_j$ .  $MP$  publishes  $\{id_i, R_i\}$ . The mask publisher  $MP$  first selects an integer  $s_0$  from the interval  $[2, N]$  and computes  $\lambda$  such that  $s_0\lambda = 1 \text{ mod } \phi(N)$ , where  $\phi(N)$  is the Euler phi-function; and then  $MP$  computes  $R_0 = g^{s_0} \text{ mod } N$ . Finally, the  $MP$  calculates  $R'_i = R_i^{s_0} \text{ mod } N$  and the mask value  $M_i = M(R'_i)$  for each evaluator  $E_i$ .  $MP$  publishes  $\{R_0, \lambda\}$ .

Weight Publisher  $WP_{(i,j)}$  enlarges its weight  $\tilde{w}(i, j)$ :  $w(i, j) = \tilde{w}(i, j) + t_w \times (j - i)$  where  $t_w$  is a threshold parameter of  $WP_{(i,j)}$ . The extension will not change the optimal solution of the longest path from  $S$  to  $n$ .  $\tilde{f}(i)$  and  $f(i)$  are denoted the original weight value  $\tilde{w}(i, j)$  and the extended weight  $w(i, j)$  of node  $i$ , respectively. Then, for each node  $j$ ,  $f(j) = \tilde{f}(j) + t_w \times (n - j)$ . So, the maximum can be computed and the secure computing is performed in Section 4.2. The polynomial  $H_{(i,j)}$  for node  $i$  is randomly chosen by weight publisher  $WP_{(i,j)}$  s.t.  $deg(H_{(i,j)}) = w(i, j)$ , and  $H_{(i,j)}(0) = c$ . The  $WP_{(i,j)}$  holds it secret.

**4.5.2 Construction phase**

The weight publisher  $WP_{(i,j)}$  performs the following steps:

- 1) Compute  $Y_i = H_{(i,j)}(R'_i) \text{ mod } N$ ;
- 2) Send  $Y_i$  to the evaluator  $E_i$ .

Each evaluator  $E_i$  computes performs the following steps to obtain the  $i$ th share of the optimal value:

1) Computes

$$F_j(R_i) = \sum_{(i,k)} (H_{(i,k)}(R_i)) \times F_k(R_i) \quad (7)$$

for  $j = n - 1, n - 2, \dots, 0$ , where  $F_j(x)$  is the optimized polynomial, which represents the longest path from the start node  $S$  to node  $j$ , and  $F_n(x) = 1$ .

2) Publishes  $HM_i = H_{(0,i)} \times F_i + M_i$ . The Eq. (7) is related to the recurrence relation of dynamic programming, as described in Eq. (1).

### 4.5.3 Recovery and verification phase

Without loss of generality, let  $E = \{E_1, E_2, \dots, E_{d+1}\}$ . The evaluators of  $E$  will recover the polynomial  $HM_i = H_{(0,i)} \times F_i + M_i$  based on following procedure.

- 1) Each evaluator calculates  $R'_i = R_0^{s_i} \pmod N$  to obtain the share, where  $s_i$  is the share of  $HM_i$ .
- 2) The evaluator in  $E$  verifies  $R'_i$ , which is provided by  $E_i$ . If  $R'_i \lambda = R_i \pmod N$ , then  $R'_i$  is legitimacy; Otherwise,  $R'_i$  is false, which means that  $E_i$  might be a cheater. The share will be discarded.
- 3) Recover the polynomial: the polynomial  $HM_i$  can be uniquely determined as follows:

$$F_j = \sum_{i=1}^{d+1} (H_{(0,i)} \times F_i + M_i) \prod_{j=1, j \neq i}^{d+1} \frac{x - R'_j}{R'_i - R'_j} = S_1 + S_2x + \dots + S_dx^d \quad (8)$$

As described in Section 4.2, evaluators check whether  $\deg(F_0) \leq d$ . Evaluators can verify whether  $F_0 = c$  or not. For instance, if  $c = 0$ ,  $F_0$  should be equal to 0. We can perform binary search to figure out the optimal value  $f(0) = \deg(F_0)$ , and publish it.

### 4.5.4 Tracing the optimal path

Evaluators calculate the optimal path as follows:

Assume that the evaluators know  $f(j) = \deg(F_j)$ , and they want to trace to node  $k$  s.t.  $\deg(F_j) = \deg(H_{(j,k)} \times F_k + M_j)$ . We test whether  $\deg(H_{(j,k)} \times F_k + M_j) = \deg(F_j) - 1$  or not for all nodes  $k$  linked to node  $j$ . The evaluators know that the node  $k$  attains  $f(j)$  when the inequality does not hold for node  $k$ . They can determine  $f(k) = \deg(F_k)$  as in Section 4.5.3 after finding the node  $k$  that attains  $f(j)$ , and publish it. Iterating this procedure recursively yields to the optimal path.

### 4.6 An example

Here, we give an example of one-dimensional graph shown in Fig. 3 to explain how to apply our scheme.

There are three links,  $(S, 1)$ ,  $(S, 2)$ , wherein weighers are  $\{2, 1, 2\}$ , respectively. The weight publishers

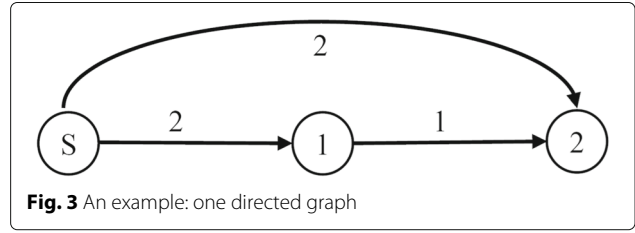


Fig. 3 An example: one directed graph

$WP_{(S,1)}$ ,  $WP_{(1,2)}$ ,  $WP_{(S,2)}$  generate the following polynomials for these links:

$$\begin{cases} H_{(S,1)} = x^2 - x \\ H_{(1,2)} = x \\ H_{(S,2)} = 2x^2 + 2x \end{cases} \quad (9)$$

There are four evaluator  $\{E_1, E_2, E_3, E_4\}$ , which randomly choose  $x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 4$ , respectively. For simplicity, we assume that  $t_w = 0$  and  $c = 0$ .

First, the mask publisher  $MP$  first chooses mask polynomial  $M(x) = x^2$ , and chooses two primes  $p = 5$  and  $q = 7$ , and calculates  $N = 5 \times 7 = 35$ . Then, the mask publisher  $MP$  chooses the generator  $g = 2$  and a randomly number  $s_0 = 5$ , and computes  $\lambda = 5$  from  $s_0\lambda = 1 \pmod{(\phi(N) = 24)}$ .  $MP$  computes  $R_0 = g^{s_0} \pmod N = 2^5 \pmod{35} = 32$ .  $MP$  publishes  $\{g, N, R_0, \lambda\}$ .

Second, the evaluator  $E_i$  computes  $R_i = g^{x_i} \pmod N$ , so four evaluators  $\{E_1, E_2, E_3, E_4\}$  generate  $R_1 = 2^1 \pmod{35} = 2, R_2 = 2^2 \pmod{35} = 4, R_3 = 2^3 \pmod{35} = 8, R_4 = 2^4 \pmod{35} = 16$ , respectively. The evaluators  $\{E_1, E_2, E_3, E_4\}$  send  $\{R_1, R_2, R_3, R_4\}$  to  $MP$  separately.

Third,  $MP$  computes  $R'_1 = R_1^5 \pmod{35} = 32, R'_2 = R_2^5 \pmod{35} = 9, R'_3 = R_3^5 \pmod{35} = 8, R'_4 = R_4^5 \pmod{35} = 11$ , and computes the mask value  $M_1 = R_1'^2 = 1024, M_2 = R_2'^2 = 81, M_3 = R_3'^2 = 64, M_4 = R_4'^2 = 121$ , and then sends  $\{\{R'_1, M_1\}, \{R'_2, M_2\}, \{R'_3, M_3\}, \{R'_4, M_4\}\}$  to evaluators  $\{E_1, E_2, E_3, E_4\}$ , respectively.

Each evaluator computes its shares following Section 4.5.2. The evaluators' corresponding computation are shown in Table 1.

When the evaluators work together to figure out the optimal result, the evaluators verify identities of participants each other using the method in Section 4.5.3 first. If all the evaluators pass the verification, from Table 1, the evaluators can recover  $F_0(x) = x^3 + x^2 + 2x$  from the shares  $F_0(32) = 33,856, F_0(9) = 828, F_0(8) = 592$ , and  $F_0(11) = 1476$ , where  $F_0(0) = 0$ . According to the Eq. (1) and (7), we figure out that  $f(0) = 3$ . The evaluators also can recover the mask polynomial  $M(x) = x^2$  according to the mask shares. Because the polynomial of degree 2, which is reconstructed from the shares of  $H_{(S,1)} \times F_1 + M$ ,

**Table 1** Each evaluator’s shares

	$H_{(S,1)}$	$H_{(1,2)}$	$H_{(S,2)}$	$F_1$	$F_0$	$M$	$H_{(S,1)} \times F_1 + M$	$H_{(S,2)} \times F_2 + M$
$E_1$	992	32	2112	32	33,856	1024	32,768	3136
$E_1$	72	9	180	9	828	81	729	261
$E_1$	56	8	144	8	592	64	512	208
$E_1$	110	11	266	11	1476	121	1331	387

does not equal to 0, the link  $(S, 1)$  attains  $f(0) = 3$ , which means that the link  $(S, 1)$  is included in the optimal result.

**5 Result and discussion**

In this section, we discuss the security properties of the proposed scheme and analyze the performance of the proposed scheme.

**5.1 Security analysis**

In this subsection, we discuss the security properties of the proposed scheme in terms of resistance against active attacks, resistance against passive attacks, non-repudiation, and accountability.

**5.1.1 Resistance against active attacks**

- **Conspiracy attacks:**In order to recover the secrets, we assume that two evaluators have a collusion activity. For example, two evaluators  $E_i$  and  $E_j$  can exchange their value  $s_i$  and  $s_j$ . So,  $E_i$  holds  $s_j$  while  $E_j$  holds  $s_i$ . Then,  $E_i$  calculates  $R_j'^{\lambda} = R_j$  while  $E_j$  computes  $R_i'^{\lambda} = R_i$ . Therefore,  $E_i$  and  $E_j$  might try to pass the verification. However, it is not impossible as the  $Id$  and  $(Id, R)$  pairs have been published by all evaluators. Thus, the conspiracy of the participants  $E_i$  and  $E_j$  can be easily recognized by other participants.
- **Evaluator cheating:** Assume that an evaluator  $E_i$  wants to gain a secret  $(s)$  via providing a false private key  $R_j$ .  $E_i$  calculates  $R_i' = R_0^{s_j}$  mod  $N$  and broadcasts it. However, other participants can check the validity of  $R_i'$  by calculating  $R_i'^{\lambda} = R_j \neq R_i$  when receiving  $R_i'$  provided by  $E_i$ . Because that the  $Id_i$  and the  $R_i$  of  $E_i$  are published, it is easy to detect that  $E_i$  provides an incorrect  $R_i'$ .
- **Reconstruct the polynomial:** Assume that an adversary  $adv$  wants to use fewer than  $t$  shares ( $t < d$ ) to reconstruct the polynomial  $HM_i$ , it is not impossible because that it equals to break Shamir’s scheme, which has been proved that it holds the security property.
- **Reveal the secret key of the evaluator:** Assume an adversary wants to obtain the participant  $E_i$ ’s secret shadow  $s_i$  from the public information  $R_i$ . He obtains  $s_i$  from  $R_i = g^{s_i}$ ; however, he has to solve the discrete logarithm problem (*DLP*), which is an *NP*-hard

problem. So, it is not impossible to obtain the secret key from the evaluator.

**5.1.2 Resistance against passive attacks**

Because that all published shares with random polynomials are masked by the mask publisher, meanwhile the extended weight  $w(i, j) = deg(H(i, j))$  is equal to or larger than  $d$ , the adversary can not obtain any information from masked shares when the number of weight publishers is less than the threshold  $d$ . Thus, the proposed scheme is secure against passive adversaries.

- **Non-repudiation:**

**Theorem 1** *If a bidder (Weight Publisher) makes a bid, it cannot deny making the bid in a later time.*

*Proof* If a bidder (Weight Publisher) make a bid, because that the evaluators work together to figure out the optimal result, and each participant is verified by other participants. If some Weight Publisher deny making the bid, the other evaluators can work together to trace all the internment mask result to verify whether the Weight publisher is lie or not according to the optimal result. □

- **Accountability:**Accountability is required to secure a system from the aspects of integrity, confidentiality, and privacy [26–30]. An accountability mechanism is

**Table 2** The properties of our scheme and the schemes proposed in [16, 17]

Properties	Our scheme	Scheme in [17]	Scheme in [16]
Resist the evaluators cheating activity	Yes	No	No
Secure channel	No	Yes	Yes
Verifiable	Yes	Yes	No
Efficient reconstruction and trace	Yes	Yes	Yes
The reusability of the secret shadow	Yes	No	No
Third party	No	Yes	Yes
Select the secret shadow by the evaluators	Yes	No	No

**Table 3** Communication complexity

Phases	The proposed scheme	Add and multiple protocol in [17]	Scheme in [16]
Initialization phase	0	$3q \times l$	$q \times l$
Construction phase	0	$q \times l$	0
Recovery and verification (recovery)	0	$d \times l \times \log l$	$d \times l \times \log l$
Tracing to the optimal path	$d \times l \times (q + \log l)$	$d \times l \times (q + \log l) + 1$	$d \times l \times (q + \log l)$

typically utilized to figure out who is responsible for what. In essence, accountability means that the system is recordable and traceable, which implies that making any entity in the system accountable for all its actions. Under such a consideration, our scheme is accountable as the evaluators can verify each other and work together to obtain the optimal result, which can be used as an evidence for dispute resolution; therefore, no one can deny its actions. Thus, we claim that the scheme has the property of accountability.

## 5.2 Performance analysis

In this section, we discuss the performance properties of our scheme and compare our schemes with others. The comparison of the properties of our scheme and the schemes proposed in [16, 17] is shown in Table 2. The details are presented as follows:

- In [16, 17], the third party is needed, which may be dishonest. Hence, the original secrets may not be reconstructed by the evaluators. In our scheme, it is impossible for the third party to cheat the evaluators as the evaluators choose their own shadows.
- The validity of the shares of each evaluator can be checked by other evaluators; the proposed scheme is *verifiable*. This improves upon [16] in which the source of the other share cannot be verified by the evaluator. If a wrong share is provided by one evaluator, which can not be figured out by other evaluators.
- In [16, 17], the shadows of the evaluators are received from the third party via secure channel; however, our scheme never discloses the shadow of each evaluator in the recovery and verification phases, and the shadow can be reused.
- In [16, 17], the secret shadows are transmitted via a private secure channel by weight publishers; however, in our scheme, the shadows are not transmitted by the weight publishers via secure channel because that the secret shadow is chosen by the evaluators themselves.

Table 3 shows round complexity during each phase. The proposed scheme does not consider communications without secure channels, i.e., the weight publisher or the evaluators publish shares in our scheme, which can be implemented by a bulletin board. Here,  $q$  is the number

of links,  $n$  is the number of nodes,  $l$  is the number of evaluators (which is equal to or greater than possible maximal value),  $d + 1$  is the number of masks, and  $N$  is the order of the finite field  $Z_N$ .

Note that our approach does have one disadvantage: if the number of nodes is very large, our scheme may be invalid sometimes because the combinatorial auction's winner determination problem is NP-complete.

## 6 Conclusions

In this paper, we presented a privacy-preserving combinatorial auction without an auctioneer scheme. In our scheme, the price is represented as the degree of a polynomial; thus, the degree of the sum/product of the two polynomials constructs the maximum/sum of the degree of two polynomials. The bidders information is hidden, and the legitimacy of the evaluator is also verified based on secret sharing, which can resist collusion attacks.

Our future research will focus on the following direction: design more efficient approaches based on greedy algorithm to protect the privacy of combinatorial auction, which would be much more suitable for practical applications.

### Acknowledgements

We are very grateful to Dr. Xiuzhen Cheng and Dr. Maya Larson who have helped improve the quality of this paper.

### Funding

This project was partially supported by the National Natural Science Foundation of China under grants 61702062, 61672119, 61472418 and 61571049, and the National Science Foundation of the USA under grants: CNS-1407986, CNS-1443858, CNS-1704397 and IIS-1741279, and the Natural Science Foundation of Chongqing (cstc2015jcyjA40037).

### Authors' contributions

All the authors developed the solution of the problem. CH proposed the main idea of the paper and finished the draft of the paper. RL, BM, and WL discussed and improved the scheme. AA and RB focused on smoothing out the language of the paper. All authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

### Author details

<sup>1</sup>Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing, China. <sup>2</sup>School of Software Engineering, Chongqing University, Chongqing, China. <sup>3</sup>Department of Computer Science, George Washington University, Washington, USA.

<sup>4</sup>Department of Computer Science, Georgia State University, Atlanta, USA.

<sup>5</sup>College of Information Science and Technology, Beijing Normal University, Beijing, China.

Received: 19 November 2017 Accepted: 30 January 2018

Published online: 12 February 2018

## References

1. K Sako, in *Proceedings of Public Key Cryptography 2000*. Universally verifiable auction protocol which hides losing bids (Springer, Melbourne, 2000), pp. 35–39
2. C Cachin, in *Proceedings of the 6th ACM Conference on Computer and Communications Security*. Efficient private bidding and auctions with an oblivious third party (ACM, Singapore, 1999), pp. 120–127
3. K Kobayashi, H Morita, K Suzuki, M Hakuta, Efficient sealed-bid auction by using one-way functions. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **84**(1), 289–294 (2001)
4. K Suzuki, K Kobayashi, H Morita, in *Information Security and Cryptology-CISC 2000*. Efficient sealed-bid auction using hash chain (Springer, Seoul, 2001), pp. 183–191
5. Q Huang, Y Tao, F Wu, in *INFOCOM, 2013 Proceedings IEEE*. Spring: A strategy-proof and privacy preserving spectrum auction mechanism (IEEE, Turin, 2013), pp. 827–835
6. M Pan, X Zhu, Y Fang, Using homomorphic encryption to secure the combinatorial spectrum auction without the trustworthy auctioneer. *Wirel. Netw.* **18**(2), 113–128 (2012)
7. M Pan, J Sun, Y Fang, Purging the back-room dealing: Secure spectrum auction leveraging paillier cryptosystem. *IEEE J. Sel. Areas Commun.* **29**(4), 866–876 (2011)
8. M Larson, R Li, C Hu, W Li, X Cheng, R Bie, in *Wireless Algorithms, Systems, and Applications*. A bidder-oriented privacy-preserving vcg auction scheme (Springer, Qufu, 2015), pp. 284–294
9. M Larson, W Li, C Hu, R Li, X Cheng, R Bie, in *Wireless Algorithms, Systems, and Applications*. A secure multi-unit sealed first-price auction mechanism (Springer, Qufu, 2015), pp. 295–304
10. W Li, M Larson, C Hu, R Li, X Cheng, R Bie, Secure multi-unit sealed first-price auction mechanisms. *Secur. Commun. Netw.* **9**(16), 3833–3843 (2016)
11. A Alrawais, A Alhothaily, J Yu, C Hu, X Cheng, Secureguard: a certificate validation system in public key infrastructure. *IEEE Trans. Veh. Technol.* (2018). Preprint
12. P Paillier, in *Advances in cryptology-EUROCRYPT'99*. Public-key cryptosystems based on composite degree residuosity classes (Springer, Prague, 1999), pp. 223–238
13. K Xing, C Hu, J Yu, X Cheng, F Zhang, Mutual privacy preserving  $k$ -means clustering in social participatory sensing. *IEEE Trans. Ind. Inform.* **13**(4), 2066–2076 (2017)
14. C Hu, X Liao, X Cheng, Verifiable multi-secret sharing based on LFSR sequences. *Theor. Comput. Sci.* **445**, 52–62 (2012)
15. H Kikuchi,  $(m+1)$  st-price auction protocol. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **85**(3), 676–683 (2002)
16. K Suzuki, M Yokoo, in *Financial Cryptography*. Secure combinatorial auctions by dynamic programming with polynomial secret sharing (Springer, Guadeloupe, 2003), pp. 44–56
17. M Nojoumian, DR Stinson, in *Information Security Practice and Experience*. Efficient sealed-bid auction protocols using verifiable secret sharing (Springer, Fuzhou, 2014), pp. 302–317
18. M Larson, C Hu, R Li, W Li, X Cheng, in *Proceedings of the 2015 Workshop on Privacy-Aware Mobile Computing*. Secure auctions without an auctioneer via verifiable secret sharing (ACM, Hangzhou, 2015), pp. 1–6
19. R Bellman, Dynamic programming and lagrange multipliers. *Proc. Natl. Acad. Sci. U. S. A.* **42**(10), 767 (1956)
20. A Shamir, How to share a secret. *Commun. ACM.* **22**(11), 612–613 (1979)
21. G Blakley, Safeguarding cryptographic keys. *Proc. Natl. Comput. Conference 1979.* **48**, 313–317 (1979)
22. C Hu, W Li, X Cheng, J Yu, S Wang, R Bie, A secure and verifiable access control scheme for big data storage in clouds. *IEEE Transactions on Big Data* (2018). Preprint
23. MH Dehkordi, S Mashhadi, An efficient threshold verifiable multi-secret sharing. *Comput. Stand. Interfaces.* **30**(3), 187–190 (2008)
24. C Hu, N Zhang, H Li, X Cheng, X Liao, Body area network security: a fuzzy attribute-based signcryption scheme. *IEEE J. Sel. Areas Commun.* **31**(9), 37–46 (2013)
25. C Hu, X Liao, D Xiao, Secret image sharing based on chaotic map and chinese remainder theorem. *Int. J. Wavelets Multiresolution Inf. Process.* **10**(03), 1250023–118 (2012)
26. J Liu, Y Xiao, J Gao, Achieving accountability in smart grid. *IEEE Syst. J.* **8**(2), 493–508 (2014)
27. R Jagadeesan, A Jeffrey, C Pitcher, J Riely, in *Computer Security—ESORICS 2009*. Towards a theory of accountability and audit (Springer, Saint-Malo, 2009), pp. 152–167
28. T Truderung, A Vogt, et al, in *Proceedings of the 17th ACM Conference on Computer and Communications Security*. Accountability: definition and relationship to verifiability (ACM, Chicago, 2010), pp. 526–535
29. J Feigenbaum, AD Jaggard, RN Wright, in *Proceedings of the 2011 Workshop on New Security Paradigms Workshop*. Towards a formal model of accountability (ACM, Marin County, 2011), pp. 45–56
30. K Ko, DA Frincke, T Goan Jr, T Heberlein, K Levitt, B Mukherjee, C Wee, in *Proceedings of the 1st ACM Conference on Computer and Communications Security*. Analysis of an algorithm for distributed recognition and accountability (ACM, Fairfax, 1993), pp. 154–164

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)