Efficient Sealed-Bid Auction Protocols Using Verifiable Secret Sharing

Mehrdad Nojoumian[1] and Douglas R. Stinson[2]

[1] Department of Computer Science
Southern Illinois University, Carbondale, Illinois, USA
nojoumian@cs.siu.edu
[2] David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, Ontario, Canada
dstinson@math.uwaterloo.ca

Abstract. This article proposes efficient solutions for the construction of sealed-bid second-price and combinatorial auction protocols in an active adversary setting. The main reason for constructing secure auction protocols is that the losing bids can be used in the future auctions as well as negotiations if they are not kept private. Our motivation is to apply *verifiable secret sharing* in order to construct various kinds of sealed-bid auctions. We initially propose two *secure second-price auction* protocols with different masking methods. Subsequently, we provide two *secure combinatorial auction* protocols based on our second masking approach. In the first scheme, we apply an existing *dynamic programming* method. In the second protocol, we use inter-agent negotiation as an approximate solution in the *multiple traveling salesman problem* to determine auction outcomes. It is worth mentioning that our protocols are independent of the secret sharing scheme that is being used.

Keywords: Applied cryptography, security and privacy in auctions.

1 Introduction

The growth of e-commerce technologies has created a remarkable opportunity for *secure auctions* where bidders submit sealed-bids to auctioneers and then the auctioneers define outcomes without revealing the losing bids. The main motivation for protection of the losing bids is that the bidders' valuations can be used in the future auctions and negotiations by different parties, say auctioneers to maximize their revenues or competitors to win the auction. This problem can be resolved by constructing privacy-preserving auction protocols.

In fact, *secure comparison*, as the main building block of sealed-bid auctions, is first motivated by the *millionaires' problem* [38]. In this problem, the goal is to determine whether $x > y$, where both x and y are private secrets of two players. The answer to this question becomes known to the parties only after the execution of the protocol. The millionaires' problem ultimately leaded to the introduction of *secure multiparty computation* MPC, where n players cooperate to perform a computation task based on the private data they each provide.

X. Huang and J. Zhou (Eds.): ISPEC 2014, LNCS 8434, pp. 302–317, 2014.

Other methods were later proposed in order to construct protocols for *secure comparison*, *interval test* and *equality test*. For instance, [9] proposes multiparty computation techniques to implement these operations. The main building block of this construction is a protocol, named *bit-decomposition*, that converts a polynomial sharing of a secret into shares of its bits. This protocol is simplified in [22]. As a counterpart, [11] implements these operations by homomorphic encryption in a computationally secure setting. We later clarify why these operations are too expensive to build practical sealed-bid auctions, as shown in Table 2.

There exist many sealed-bid auctions in both *passive* and *active* adversary models. In the former, players follow protocols correctly but are curious to learn private bids. In the latter, players may also deviate from protocols. The majority of the sealed-bid auction protocols either are secure only in the passive adversary model or they apply costly bitwise approaches, e.g., using *verifiable secret sharing* VSS for every single bit of each bid rather than a single VSS for the entire bid.

1.1 Literature Review

All the following protocols utilize "secret sharing" as their main building block. In the initial construction of the first-price sealed-bid auction protocols, the authors in [10] implement a secure auction service by using verifiable secret sharing as well as verifiable signature sharing. At the end of the bidding time, auctioneers open bids to define outcomes, therefore, they learn the losing bids.

The authors in [12] illustrate a set of protocols for sealed-bid auctions by using secure distributed computation. The bidders' valuations are never revealed to any party even when the auction is completed. Their constructions support the first-price and second-price auctions. The general idea of their approach is to compare bids digit-by-digit by applying secret sharing techniques. This protocol is computationally very expensive.

The proposed first-price construction in [14] (modified in [15]) demonstrates a multi-round secure auction protocol in which winners from an auction round take part in a subsequent tie-breaking second auction round. The authors use the addition operation of secure multiparty computation in a passive adversary model. Later, the authors in [27] detected some shortcomings in this scheme such as the lack of verifiability. They then resolved those problems.

The authors in [13] present a protocol for the $(M+1)$st-price auction. They illustrate a new method where bidders' valuations are encoded by the degree of distributed polynomials. The proposed construction requires only two rounds of computations; one round for bidders and one round for auctioneers. The proposed scheme in [6] is a fully private $(M+1)$st-price auction protocol in which only the winning bidders and the seller learn the selling price. It has two main shortcomings. First, the scheme is not able to handle ties among multiple winners. Second, it is not an efficient construction in the computational setting.

Finally, the authors in [26] design a new first-price secure auction protocol based on a homomorphic secret sharing scheme. Their construction relies on hard computation problems and does not depend on any trust. They also show that the proposed protocol is secure against different kinds of attacks.

1.2 Motivation and Contributions

Our motivation is to propose efficient solutions for the construction of secure second-price as well as combinatorial auction protocols where losing bids are kept private in an active adversary model. We would like to use secret sharing techniques to define auction outcomes without using costly bitwise operations. This helps us to design approximate secure solutions for the general combinatorial auction that is expensive even without using sealed-bids. Even by having unlimited computational power, one can significantly reduce the communication cost by avoiding the use of bitwise operations, that is, sharing each bid as a single field element is more efficient compared to sharing every single bit/digit of that element, for an example of such a scheme, see [12]. Our constructions consist of an initializer $\mathcal{I}$, n bidders and m auctioneers. Here are our contributions:

- Our first solution is proposed for the second-price auction by using VSS. In this protocol, all bids are masked by using $+$ operation, consequently, bids are sealed but their differences are revealed only to auctioneers. Although the general idea is similar to the comparison protocol in [22], our protocol works in an active adversary setting without using any bitwise operation. In that article, the authors use bitwise operations in a passive adversary model.
- We then improve our previous solution in order to prevent the revelation of the difference between each pair of bids. We propose another sealed-bid second-price auction protocol where all bids are masked by using $+$ and $\times$ operations. As a result, both bids and their differences are sealed, however, the ratio of the bids are revealed only to auctioneers. We should stress that this protocol can be simply extended to the secure $(M+1)$st-price auction.
- Finally, we provide two secure combinatorial auction protocols based on our second masking approach: (a) we use an existing *dynamic programming* method [33] to define auction outcomes. In that paper, the authors encode bids as the degree of secret sharing polynomials. As a result, their protocol only works in the passive adversary model whereas our construction works in an active adversary setting, (b) we apply the inter-agent negotiation approach, introduced in [16], as an approximate solution in a *multiple traveling salesman problem* in order to determine auction outcomes.

2 Preliminaries

2.1 Auction Protocols

In an auction, winner is a bidder who has submitted the highest bid. To define the selling price, there are two main approaches: *first-price* and *second-price* auctions. In the former, the winner pays the amount that he has proposed, i.e., highest bid. In the latter, the winner pays the amount of the second-highest bid. There exist other types of auctions such as $(M+1)$*st-price* and *combinatorial auctions*. In the former, the highest M bidders win the auction and pay a uniform price defined by $(M + 1)$-price. In the latter, multiple items with interdependent values are sold while bidders can bid on any combination of items.

In the first-price auction, a bidder potentially is able to define the winner as well as the selling price at the same time. On the other hand, in the second-price auction, a bidder potentially is able to define either the winner or the selling price for the winner. As a result, he proposes the actual highest value, say κ, he can afford to pay, which is also a profitable price for him [35]. Suppose the proposed bid is less than κ. In this case, the bidder decreases his chance of winning. If the proposed bid is bigger than κ, the bidder might win with an unprofitable price. This property forces the bidders to propose their true valuations.

2.2 Secret Sharing

In *secret sharing schemes*, a secret is divided into various shares in order to be distributed among participants, then a subset of players cooperate to reveal the secret [31,5]. In a (t, n)-*secret sharing scheme* where $t < n$, the secret is divided into n shares such that any $t + 1$ players can combine their shares to reveal the secret, but any subset of t parties cannot learn anything about the secret.

In *verifiable secret sharing* VSS [8], players can verify the consistency of their shares with other players' shares. There are various techniques for performing the verification procedure, such as using zero knowledge proof with small probability of error, or applying bivariate polynomials without any probability of errors [4]; the latter construction works under the assumption that $n \geq 3t+1$ by considering secure pairwise channels among players. The proposed scheme in [29] applies the same communication model along with a broadcast channel to construct a new scheme with $n \geq 2t + 1$. The authors in [32] construct a VSS protocol based on symmetric bivariate polynomials. This construction is simple and uses both pairwise channels and a broadcast channel under the assumption that $n \geq 4t+1$.

3 Our Constructions

Our model, Figure 1, consists of n bidders $\mathcal{B}_1 \ldots \mathcal{B}_n$, m auctioneers $\mathcal{A}_1 \ldots \mathcal{A}_m$ and a seller. We consider communication model of VSS that is being used [32].

In addition, a trusted initializer $\mathcal{I}$ distributes some information and then leaves the scheme before our protocols start. This is preferable to a trusted party who remains in the scheme. In the literature, trusted authorities are assumed in many secure auction protocols, for instance, semi-trusted third party [7,2],

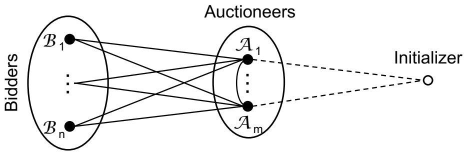

Fig. 1. Proposed Secure Auction Model

trusted third party [37,21], trusted centers [30], trusted authority [1,34], trustee [36]. It is worth mentioning that by paying an extra computational cost, a trusted party or initializer can be removed from any scheme to be replaced by MPC. Since each bidder acts as an independent dealer and the auctioneers perform the computation, our protocols have the following properties and assumptions:

- They can tolerate colluding auctioneers ∇ where $m \geq 4t+1$ and $t \geq |\nabla|$ due to VSS of [32]. If the complicated VSS of [29] is used, our protocols can tolerate $m \geq 2t+1$ since these protocols are independent of their VSS.
- They can tolerate dishonest bidders who submit inconsistent shares. Note that we assume the majority of the bidders are honest.
- We assume bidders do not collude with auctioneers similar to [25], and multi-auctioneer and multi-bidder protocols in [10,14].

There also exist other kinds of collusion assumptions in the literature, e.g., [20] assumes auctioneers do not collude with the auction issuer, [18] assumes the seller does not collude with the auction authority, etc. In Table 1, we have listed some protocols that have an assumption similar to our constructions.

Table 1. Protocols Where the Auctioneers Cannot Collude With the Bidders

Protocol	**Cryptographic Technique**	**Adversary Model**
Here	Verifiable Secret Sharing	Active
[25]	Homomorphic Encryption	Active
[14,15]	Secret Sharing	Passive
[10]	Verifiable Secret and Signature Sharing	Active

To construct our sealed-bid auction protocols, we use (1) + operation for adding two shared secrets, (2) × operation for multiplying two shared secrets. Although any arbitrary VSS can be used in our constructions, we apply the verifiable secret sharing scheme proposed in [32] due to its simplicity. This means our protocols would tolerate more dishonest auctioneers if the VSS of [4,29] were used. All computations are performed in a large enough finite field $\mathbb{Z}_q$.

3.1 Sealed-Bid Second-Price Auction Protocol Using +

In our first construction, bidders initially distribute shares of their bids β_i among auctioneers by VSS. Auctioneers then mask all shared secrets by adding an unknown value δ to bids, i.e., computing $\beta_i + \delta$ for $1 \leq i \leq n$: this increases valuations equally in order to preserve the ordering. Finally, auctioneers reveal the masked values to determine auction outcomes without revealing actual bids.

We assume that the total number of colluding auctioneers is limited to our secret sharing threshold, i.e., $|\nabla| \leq t$. We also select a large enough finite field to prevent the modular reduction after using the addition operation. Our first solution is shown in Figures 2 and 3. The first phase is repeated n times, i.e., it is used for each bidder $\mathcal{B}_k$ where $1 \leq k \leq n$.

Initialization

1. $\mathcal{I}$ initiates a secret sharing scheme by a symmetric polynomial, that is, he generates $h(x, y) \in \mathbb{Z}_q[x, y]$ of degree t in which $h(0, 0) = \delta$, where $q > \kappa + \delta$ to prevent the modular reduction and κ denotes the maximum possible price.
2. He then sends shares of auctioneers $\mathcal{A}_i$ for $1 \leq i \leq m$ accordingly and leaves the scheme, i.e., $h_i(x) = h(x, \omega^i)$ where ω is a primitive element in the field. Now, each auctioneer has a share of an unknown value δ.

Bid Submission

1. Each bidder $\mathcal{B}_k$ chooses a random symmetric polynomial $g_k(x, y) \in \mathbb{Z}_q[x, y]$ of degree t to send shares $g_{ki}(x) = g_k(x, \omega^i)$ to $\mathcal{A}_i$ for $1 \leq i \leq m$ through a private channel where $g_k(0, 0) = \beta_k$.
2. To verify distributed shares, auctioneers $\mathcal{A}_i$ and $\mathcal{A}_j$ perform pairwise checks, i.e., they verify that $g_{ki}(\omega^j) = g_{kj}(\omega^i)$, similar to VSS of [32]. They will either accept shares of β_k or disqualify $\mathcal{B}_k$.

Fig. 2. A. Secure Auction Protocol Using Addition Operation

Outcome Computation

1. Each auctioneer $\mathcal{A}_i$ **locally** adds $h_i(x)$ to the share that he has received from each bidder $\mathcal{B}_k$, that is, $\psi_{ki}(x) = g_{ki}(x) + h_i(x)$ for $1 \leq k \leq n$. In fact, $\psi_{ki}(x)$ are shares of $\beta_k + \delta$ for $1 \leq k \leq n$ where δ is unknown to everyone.
2. Each $\mathcal{A}_i$ then sends $\psi_{ki}(0)$ to a selected auctioneer $\mathcal{A}_j$ where $i, j \in \mathit{\Gamma}$, i.e., the set of good auctioneers. All computations performed by $\mathcal{A}_j$ are **only** visible to the auctioneers. $\mathcal{A}_j$ computes $\varphi_k(0, y)$ such that $\varphi_k(0, \omega^i) = \psi_{ki}(0)$ for at least $m - 2|\nabla|$ values of i.
3. In fact, $\varphi_k(x, y) = g_k + h$. $\mathcal{A}_j$ computes masked values $\varphi_k(0, 0) = \beta_k + \delta$ and then sorts them in decreasing order, i.e., $\varphi^1_w(0, 0), \varphi^2_s(0, 0), \ldots, \varphi^n_*(0, 0)$, where w is the index of the winner and s is the index of the second highest bid.
4. Auctioneers send the winner's index along with shares $g_{si}(x)$-s to all bidders through private channels. Each bidder **locally** computes the selling price by $g_s(0, 0) = \beta_s$. They can agree on β_s due to the honest majority assumption.

Fig. 3. B. Secure Auction Protocol Using Addition Operation

Since equal bids have equal masked values, ties among multiple winners can be detected and handled by assigning priority to bidders or by a random selection.

Theorem 1. *The proposed protocol defines auction outcomes correctly and only reveals the difference between each pair of bids to the auctioneers in an active adversary setting. We require $m \geq 4t+1$ even if one bidder is dishonest, otherwise, we require $m \geq 3t+1$ when we use VSS of [32].*

Proof. The security of the verifiable secret sharing scheme that we use is proven in [32]. We provide further clarifications on the condition of this construction. Dishonest auctioneers have two possibilities: (a) they either collude to recover secret bids or (b) they send incorrect shares to disrupt the protocol.
(a) t**-privacy:** If all colluding auctioneers $|\nabla| \leq t$ collect their shares, they are not able to recover secret bids β_k since all secret sharing polynomials $g_k(x, y)$'s are of degree t and each of which requires $t+1$ shares to be interpolated.
(b) t**-resilience:** On the other hand, dishonest auctioneers cannot disrupt the protocol. In the worst case scenario, if a dishonest bidder sends incorrect shares (i.e., less than $\frac{1}{4}$ of shares can be corrupted for an acceptable bid submission) to honest auctioneers during *bid submission* and also colluding auctioneers send incorrect shares (i.e., less than $\frac{1}{4}$ of the remaining $\frac{3}{4}$ shares) to the selected auctioneer $\mathcal{A}_j$ for the reconstruction of $\varphi_k(0, 0)$ in the *outcome computation* phase, $\mathcal{A}_j$ can then use an error correction technique, such as the Reed-Solomon Codes [19], to interpolate $\varphi_k(0, y)$. Finally, if dishonest auctioneers $|\nabla| \leq t$ send incorrect shares to bidders in the step-4 of the *outcome computation* phase, they can each use error correction to recover β_s. If all bidders are honest, $m \geq 3t+1$ satisfies the required condition of error correction.

Note that the dishonest bidders cannot disrupt the protocol if they collude with each others because they are only involved in two tasks. (a) Bid submission by VSS: either honest auctioneers receive consistent shares with respect to a secret bid and accept secret sharing, or the bidder is disqualified. In the former case, the bidder cannot repudiate his bid since those consistent shares are a strong commitment. (b) Selling price reconstruction: each bidder receives the winner's index along with shares of the selling price from auctioneers in order to compute the outcome, therefore, colluding bidders cannot disrupt the protocol since majority of bidders are honest and they are able to agree on the winner's index and a correct selling price.

At the end of the protocol, all bidders only know auction outcomes. Assuming bidders do not collude with auctioneers, all losing bids are kept secret from all parties because δ is an anonymous constant term only known to $\mathcal{I}$. It is worth mentioning that revealing $\varphi_k(0, 0)$'s only discloses the masked values $\beta_k + \delta$ and the difference of each pair of bids to auctioneers, but not the actual bids β_k. □

3.2 Sealed-Bid Second-Price Auction Protocol Using × and +

We now apply a practical approach to hide bids as well as their exact distances. Similar to the previous approach, bidders initially distribute shares of their bids among auctioneers by VSS. Then, auctioneers mask shared secrets to define outcomes. They start by comparing each pair of consecutive bids from β_1 all the way to β_n to find the maximum element and repeat this process to define the second maximum element, i.e., $(n-1)+(n-2) = 2n-3$ comparisons in total.

For each comparison, they multiply two bids by a new unknown secret $1 < \alpha_l$ and then add two new random secrets δ_{l1} and δ_{l2} (as noise) to the resulting values such that the order of two bids are maintained, i.e., $\alpha_l\beta_k + \delta_{l1}$ and $\alpha_l\beta_{k+1} + \delta_{l2}$ where $1 \leq \delta_{l1} \neq \delta_{l2} < \alpha_l$, shown in Figures 4 and 5. Each time, after executing × operation, the degree reduction protocol in [24] is used to adjust the threshold.

Initialization

1. Initializer $\mathcal{I}$ generates symmetric polynomials $f_l, h_{l1}, h_{l2} \in \mathbb{Z}_q[x, y]$ of degree t for $1 \leq l \leq (2n - 3)$ with constant terms $\alpha_l, \delta_{l1}, \delta_{l2}$, where $q > \alpha_l(\kappa + 1)$ to prevent the modular reduction. In fact, we use different l for each comparison.
2. He then sends shares of f_l, h_{l1}, h_{l2} to $\mathcal{A}_i$ for $1 \leq i \leq m$ and leaves the scheme. That is, $f_l^i(x) = f_l(x, \omega^i)$, $h_{l1}^i(x) = h_{l1}(x, \omega^i)$ and $h_{l2}^i(x) = h_{l2}(x, \omega^i)$ where ω is a primitive element.

Bid Submission

- We apply the *bid submission* protocol of the previous construction, that is, each bidder $\mathcal{B}_k$ chooses a random symmetric polynomial $g_k(x, y)$ of degree t to send $g_{ki}(x) = g_k(x, \omega^i)$ to $\mathcal{A}_i$ for $1 \leq i \leq m$ through private channels such that $g_k(0, 0) = \beta_k$. Auctioneers also verify shares similar to that protocol.

Fig. 4. Secure Auction Using Addition and Multiplication Operations

In this protocol, equal bids are not distinguished due to the random noise δ_{l1} and δ_{l2}. Therefore, auctioneers can execute a secure equality test on β_w and β_k to detect potential ties between them, i.e., compute $\gamma_l(\beta_w - \beta_k)$ where γ_l is unknown. If it is zero, two bids are equal, otherwise, they are different.

Theorem 2. *The proposed protocol defines auction outcomes correctly and only reveals the ratio of bids to the auctioneers in an active adversary setting. We require $m \geq 4t + 1$ even if a single bidder is dishonest, otherwise, we only require $m \geq 3t + 1$ when we use VSS of [32].*

Proof. We provide a short clarification since the security proof is straightforward and the same as the previous theorem. In this protocol, the actual difference of each pair of bids are kept private due to the additive factors δ_{l1} and δ_{l2}, where $1 \leq \delta_{l1} \neq \delta_{l2} < \alpha_l$. In other words, considering two consequtive bids $\beta_k < \beta_{k+1}$ where $\beta_{k+1} - \beta_k = 1$, their corresponding masked values have the same ordering, that is, $\alpha_l\beta_k + \delta_{l1} < \alpha_l\beta_{k+1} + \delta_{l2}$ even if $\delta_{l1} = \alpha_l - 1$ and $\delta_{l2} = 1$. However, upper and lower bounds of the bids' ratios are revealed only to auctioneers:

$$ratio = \frac{\alpha_l\beta_k + \delta_{l1}}{\alpha_l\beta_{k+1} + \delta_{l2}} < \frac{\alpha_l\beta_k + \alpha_l}{\alpha_l\beta_{k+1}} < \frac{\beta_k + 1}{\beta_{k+1}}$$

$$ratio = \frac{\alpha_l\beta_k + \delta_{l1}}{\alpha_l\beta_{k+1} + \delta_{l2}} > \frac{\alpha_l\beta_k}{\alpha_l\beta_{k+1} + \alpha_l} > \frac{\beta_k}{\beta_{k+1} + 1}$$

Outcome Computation

1. Auctioneers select shares of a pair of bids β_k and $\beta_{k'}$. Each $\mathcal{A}_i$ **locally** computes $\psi_{ki}(x) = f_l^i(x) \times g_{ki}(x)$ and $\psi_{k'i}(x) = f_l^i(x) \times g_{k'i}(x)$, that is, shares of new symmetric polynomials. They execute a degree reduction protocol for threshold adjustment [24]. After that, each $\mathcal{A}_i$ adds shares $h_{l1}^i(x)$ and $h_{l2}^i(x)$ to the previous shares, i.e., $\psi_{ki}(x) \leftarrow \psi_{ki}(x) + h_{l1}^i(x)$ and $\psi_{k'i}(x) \leftarrow \psi_{k'i}(x) + h_{l2}^i(x)$.
2. Each $\mathcal{A}_i$ then sends $\psi_{ki}(0)$ and $\psi_{k'i}(0)$ to a selected $\mathcal{A}_j$ where $i, j \in \Gamma$, i.e., the set of good auctioneers. Computations performed by $\mathcal{A}_j$ are **only** visible to the auctioneers. $\mathcal{A}_j$ computes polynomials $\varphi_k(0, y)$ and $\varphi_{k'}(0, y)$ such that $\varphi_k(0, \omega^i) = \psi_{ki}(0)$ and $\varphi_{k'}(0, \omega^i) = \psi_{k'i}(0)$ for at least $m - 2|\nabla|$ values of i.
3. In fact, $\varphi_k(x, y) = f_l \times g_k + h_{l1}$ and $\varphi_{k'}(x, y) = f_l \times g_{k'} + h_{l2}$. Therefore, $\mathcal{A}_j$ reveals $\varphi_k(0, 0) = \alpha_l\beta_k + \delta_{l1}$ and $\varphi_{k'}(0, 0) = \alpha_l\beta_{k'} + \delta_{l2}$. Now, auctioneers can define which bid is larger by comparing masked values $\alpha_l\beta_k + \delta_{l1}$ and $\alpha_l\beta_{k'} + \delta_{l2}$, where $\alpha_l, \delta_{l1}, \delta_{l2}$ are unknown to everyone.
4. Auctioneers repeat steps $1-3$ to determine two highest bids β_w and β_s accordingly. The winner's *id* along with shares $g_{si}(x)$-s are sent to all bidders through private channels. Each $\mathcal{B}_k$ **locally** computes the selling price by $g_s(0, 0) = \beta_s$. They can agree on β_s due to the honest majority assumption.

Fig. 5. Secure Auction Using Addition and Multiplication Operations

It is now easy to observe that the ratio of two bids are bounded as follows:

$$\frac{\beta_k}{\beta_{k+1} + 1} < ratio < \frac{\beta_k + 1}{\beta_{k+1}}$$

During the *outcome computation* phase and the execution of a degree reduction protocol, auctioneers can verify all computations by means of pairwise checks (similar to VSS of [32]) to make sure everyone is following the protocols correctly since all polynomials remain symmetric. We should mention that the degree reduction is avoidable in some other settings, e.g., having honest bidders under $m \geq 4t+1$ assumption, auctioneers can interpolate a polynomial of degree $2t$ in the existence of t malicious parties by using error correction [19]. □

3.3 Sealed-Bid Combinatorial Auction Protocol by Dynamic Programming

The first unconditionally secure combinatorial auction protocol was proposed in [33]. (For other type of unconditionally secure auction protocols see [23,17], i.e., sealed-bid Dutch-style auctions.) [33] applies a *dynamic programming* DP technique to determine auction outcomes. This solution is secure only in the passive adversary model and it is not verifiable. In addition, the number of auctioneers must be larger than the maximum possible revenue.

In this construction, weight publishers (bidders) submit their valuations as the degree of secret sharing polynomials. Then, evaluators (auctioneers) use $deg(g_k) + deg(g_l) = deg(g_k \times g_l)$ and $\max\{deg(g_k), deg(g_l)\} = deg(g_k + g_l)$ to implement *addition* and *max* operations accordingly. They also use mask publishers (trusted third parties) to execute these operations securely. The authors later propose the counterpart construction of this scheme in a computational setting based on the homomorphic encryption [39].

We first provide an example to show a combinatorial auction model based on a directed graph, Figure 6. We then illustrate the *dynamic programming* method in order to define auction outcomes. Finally, we explain our secure solution to this problem in an active adversary setting.

Example 1. Suppose six bidders $\mathcal{B}_1, \ldots, \mathcal{B}_6$ propose their evaluations on various subsets of three items $\{a, b, c\}$. For instance, $\beta_5 = \$5$ for all three items, $\beta_2 = \$3$ for $\{a\}$, and so on. As you can see, auctioneers earn the maximum revenue if they sell $\{b\}$ to the first bidder for \$1 and $\{a, c\}$ to the last bidder for \$5.

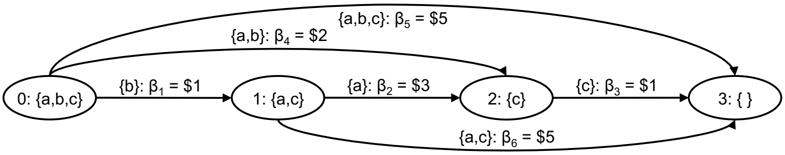

Fig. 6. Directed Graph to Model Combinatorial Auctions

$$\begin{aligned}
r &= 3 : f(3) = 0 \text{ destination function}\\
r &= 2 : f(2) = \max\{w_{23} + f(3)\} = \max\{1\} = 1\\
r &= 1 : f(1) = \max\{w_{12} + f(2), w_{13} + f(3)\} = \max\{4, 5\} = 5\\
r &= 0 : f(0) = \max\{w_{01} + f(1), w_{02} + f(2), w_{03} + f(3)\} = \max\{6, 3, 5\} = 6
\end{aligned}$$

More generally, $f(r) = \overset{link:r \to s}{\max}\{w_{rs} + f(s)\}$, where the value of the destination function is zero and w_{rs} is the weight of the link between two subsequent nodes r and s. Therefore, we need two operations *addition* and *max* in order to implement a sealed-bid combinatorial auction protocol in the active adversary setting.

Similar to the previous construction, $\mathcal{I}$ first distributes some multiplicative and additive factors based on the size of the directed graph. Bidders $\mathcal{B}_1, \ldots, \mathcal{B}_n$ then distribute their bids by symmetric bivariate polynomials of degree t. In the computation stage, auctioneers $\mathcal{A}_1, \ldots, \mathcal{A}_m$ use the same addition operation to execute $w_{rs} + f(s) = \beta_k + f(s)$. They also apply the previous comparison approach to implement the *max* operation. Finally, they define auction outcomes.

3.4 Sealed-Bid Combinatorial Auction Protocol by Multiple-TSP

In our last construction, we design an approximate secure solution in order to solve the combinatorial auction problem through a *multiple traveling salesman problem* MTSP, where more than one salesman is allowed to be used for finding the solution. In the fixed destination version of this problem, each salesman returns to his original depot after completing the tour. Similar to the traveling salesman problem, each city is visited exactly once and the total cost of visiting cities is minimized [3]. We first illustrate how to model a combinatorial auction based on the multiple traveling salesman problem and then we demonstrate an *inter-agent negotiation approach* [16] to solve this problem, Figure 7.

Example 2. Suppose three bidders $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ propose their bids on various subsets of seven items $\{a, b, c, d, e, f, g\}$, as shown below.

$$\begin{array}{lllll}
\mathcal{B}_1 \rightarrow \{a,b,c\} : \$12 & or & \{a,b\} : \$5 & or & \{a,c\} : \$7 \\
\mathcal{B}_2 \rightarrow \{d,e\} : \$7 & or & \{b,d,e\} : \$13 & or & \{c,d,e\} : \$11 \\
\mathcal{B}_3 \rightarrow \{f,g\} : \$9 & or & \{b,f,g\} : \$16 & or & \{c,f,g\} : \$14
\end{array}$$

In the initialization phase, auctioneers assign all items to three bidders for the total price of \$28, Figure 7 left-hand side. In the subsequent negotiation stages, they maximize the selling price. For instance, both items $\{b\}$ and $\{c\}$ can be release from the first bidder's set. Since $\mathcal{B}_1$ pays more money for $\{a, c\}$ compared to $\{a, b\}$, therefore $\{b\}$ is released with $12 - 7 = \$5$ cost. On the other hand, $\mathcal{B}_2$ pays extra $13 - 7 = \$6 > \$5$ for $\{b\}$ while $\mathcal{B}_3$ pays extra $16 - 9 = \$7 > \$5$ for that. Therefore, $\{b\}$ is assigned to the last bidder and the total selling price is increased to \$30 through one round of negotiation, Figure 7 right-hand side.

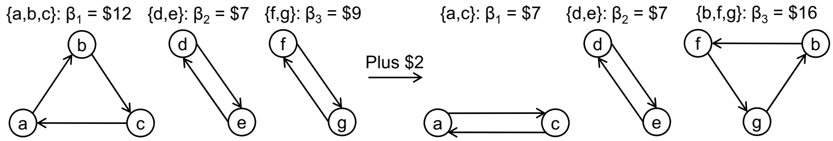

Fig. 7. Multiple Traveling Salesman Problem for Modeling Combinatorial Auctions

Similar to our previous construction, we require *addition* (minus is the same) and *max* (or comparison) operations to implement the negotiation protocol in a secure setting. We can also define a time interval for the entire protocol in order to limit the number of negotiation rounds for an approximate solution.

4 Complexity and Properties

We now clarify why the existing bitwise operations are too expensive to construct sealed-bid auction protocols. As we stated earlier, these protocols use VSS for every single bit of each bid rather than a single VSS for each bid. Let $\ell = \lceil \log_2 q \rceil$ denotes the number of bits of each bid, i.e., the size of each finite field's element.

The *round complexity* is measured by the number of rounds in which players execute the multiplication protocol and the *communication complexity* is measured by the number of invocations of the multiplication protocol. For instance, to compute $\alpha_1\alpha_2\alpha_3\alpha_4$, $\alpha_1\alpha_2$ and $(\alpha_1\alpha_2)\alpha_3$ and $(\alpha_1\alpha_2\alpha_3)\alpha_4$ can be computed sequentially, or $\alpha_1\alpha_2$ and $\alpha_3\alpha_4$ can be computed in parallel in order to compute $(\alpha_1\alpha_2)(\alpha_3\alpha_4)$. The former method takes 3 rounds with 3 invocations whereas the later method takes 2 rounds with 3 invocations. In all complexity analyses, the goal is to perform parallel multiplications as much as possible.

To construct bitwise operations, *Bit-Decomposition* BD protocol is proposed in [9] to convert a polynomial sharing of a secret into shares of its bits. This protocol takes 114 rounds and $118\ell + 110\,\ell\log\ell$ invocations. The authors also provide a protocol, named *Bitwise Less-Than* BIT-LT, to compare two decomposed elements in 19 rounds with 22ℓ invocations. Therefore, to compare two elements, they must be decomposed in parallel and then they can be compared, i.e., $(114 + 19)$ rounds and $(2 * (118\ell + 110\,\ell\log\ell) + 22\ell)$ invocations.

The authors in [22] show that the comparison protocol can be simplified by using simpler subprotocols, i.e., $(38 + 6)$ rounds and $(2 * (93\ell + 94\,\ell\log\ell) + 19\ell)$ invocations. They also show that the BD protocol itself can be simplified to achieve even a better result, i.e., $(25 + 6)$ rounds and $(2 * (93\ell + 47\,\ell\log\ell) + 19\ell)$ invocations. Finally, they propose a new comparison protocol without applying the DB protocol while using other bitwise operations. This construction takes $(13 + 2)$ rounds and $(3 * (93\ell + 1) + 2)$ invocations. Note that our comparison protocol only takes 1 round for two multiplications in parallel and 2 invocations.

The summary of these analyses are presented in Table 2. Even by using elements with $\ell = 128$ bits, it is impractical to use any of these bitwise operations. For instance, in the best case scenario, it requires $35,717$ secure multiplications in order to perform one single comparison. Having only 10 bids, it requires $350,717$ secure multiplications to find the highest bid whereas our protocol only requires 20 multiplications. This implies that avoiding bitwise operations is better than revealing partial information like ratio of bids.

Table 2. Single *Comparison*'s Cost in Terms of the Number of Multiplications

Secure Comparison Protocol	**Number of Rounds**	**Communication Complexity**	$\ell = 128$**: Number of Secure Multiplications**
Our 2nd Protocol	1	2	2
[22] not using BD	15	$279\ell + 5$	$35,717$
[22] using BD	31	$205\ell + 94\,\ell\log\ell$	$110,464$
[22] simplifying [9]	44	$205\ell + 188\,\ell\log\ell$	$194,688$
[9]	133	$258\ell + 220\,\ell\log\ell$	$230,144$

In general, sealed-bid auction protocols have some essential properties [28] as listed below. (a) *Correctness*: determining auction outcomes correctly, i.e., the winners and the selling price. (b) *Privacy*: preserving privacy of the losing bids. (c) *Verifiability*: parties who exchange money such as bidders and the seller (if applicable) must be able to verify auction outcomes. (d) *Fairness*: bidders must not be able to modify and/or deny (a.k.a non-repudiation) the submitted bids.

(e) *Robustness*: none of the active parties are assumed to be honest and malicious behavior must be tolerated. (f) *Anonymity*: the identities of the losers must be kept secret. Excluding property (f), our protocols have all the above features.

We also would like to highlight some important points regarding our protocols. Although partial information like the ratio of the bids might be revealed to auctioneers, our protocols keep the actual values of the losing bids secret. This is much better than using impractical bitwise approach to fully hide the losing bids. Moreover, revealing the ratio of the bids is better than revealing the exact difference between two bids, i.e., saying the 2nd-highest bid is 3/4 of the winning bid or saying the 2nd-highest bid is exactly $10 less than the winning bid. In our schemes, auctioneers perform similar to an intermediate computation engine. In other words, bidders determine the actual value of the selling price themselves by outsourcing part of the computation, as in the client-server MPC model. Finally, our initializer, who can be replaced by MPC, is not an active party when the auction starts.

5 Concluding Remarks

We initially illustrated the lack of efficient solutions for the sealed-bid auction protocols that are secure in an active adversary setting (without using costly bitwise operations). We therefore proposed four secure constructions with different properties and applications. The summary of our contributions are presented in Table 3. Note that $m \geq 2t + 1$ can be tolerated using complicated VSS of [29].

Table 3. Sealed-Bid Auction Protocols Using VSS of [32]

Protocol	**Adv.**	$\mathcal{A}_{1..m}$	$\mathcal{B}_{1..n}$	**Assumption**	**Opt.**	**Reveal**
2nd-price	active	dishonest	honest or dishonest	$m \geq 3t + 1$ or $m \geq 4t + 1$	$+$	bids' differences
2nd-price combinatorial DP combinatorial MTSP					$+, \times$	ratio of bids

It is quite challenging to construct sealed-bid auction protocols in an active adversary model without using a trusted party. In other words, if one relaxes these assumptions, he can decrease the computation and communication complexities. For instance, constructing the proposed schemes by considering the simple passive adversary model or using a trusted authority who remains in the scheme while the protocol is being executed. In addition, the winner determination problem of a general combinatorial auction is NP-complete and implementing this problem in a secure fashion adds extra computational cost to the protocol. Therefore, it is reasonable to apply simpler protocols (compared to bitwise approach where every single bit of bids is shared) along with approximate solutions to define auction outcomes. In this case, even by having unlimited computational power, we can significantly improve the communication cost.

References

1. Abe, M., Suzuki, K.: M+1-st price auction using homomorphic encryption. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 115–124. Springer, Heidelberg (2002)
2. Baudron, O., Stern, J.: Non-interactive private auctions. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 354–377. Springer, Heidelberg (2002)
3. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34(3), 209–219 (2006)
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: 20th Annual ACM Symposium on Theory of Computing, STOC, pp. 1–10 (1988)
5. Blakley, G.R.: Safeguarding cryptographic keys. In: National Computer Conference, vol. 48, pp. 313–317. AFIPS Press (1979)
6. Brandt, F.: A verifiable, bidder-resolved auction protocol. In: 5th Int. Workshop on Deception, Fraud and Trust in Agent Societies, Special Track on Privacy and Protection with Multi-Agent Systems, pp. 18–25 (2002)
7. Cachin, C.: Efficient private bidding and auctions with an oblivious third party. In: ACM Conference on Computer and Communications Security, pp. 120–127 (1999)
8. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: 26th IEEE Annual Symposium on Foundations of Computer Science, FOCS, pp. 383–395 (1985)
9. Damgård, I.B., Fitzi, M., Kiltz, E., Nielsen, J.B., Toft, T.: Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 285–304. Springer, Heidelberg (2006)
10. Franklin, M.K., Reiter, M.K.: The design and implementation of a secure auction service. IEEE Transactions on Software Eng. 22(5), 302–312 (1996)
11. Garay, J.A., Schoenmakers, B., Villegas, J.: Practical and secure solutions for integer comparison. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 330–342. Springer, Heidelberg (2007)
12. Harkavy, M., Tygar, J.D., Kikuchi, H.: Electronic auctions with private bids. In: 3rd Workshop on E-Commerce, pp. 61–74. USENIX Association (1998)
13. Kikuchi, H.: (M+1)st-price auction protocol. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 341–363. Springer, Heidelberg (2002)
14. Kikuchi, H., Harkavy, M., Tygar, J.D.: Multi-round anonymous auction protocols. IEICE Transaction on Information and Systems 82, 769–777 (1999)
15. Kikuchi, H., Hotta, S., Abe, K., Nakanishi, S.: Distributed auction servers resolving winner and winning bid without revealing privacy of bids. In: 7th Int. Conference on Parallel and Distributed Systems, pp. 307–312. IEEE (2000)
16. Kim, I.C.: Task reallocation in multiagent systems based on vickrey auctioning. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 40–44. IOS Press (2002)
17. Krishnamachari, S., Nojoumian, M., Akkaya, K.: Implementation and analysis of dutch-style sealed-bid auctions: Computational vs unconditional security (2014) (Under Review Manuscript)
18. Lipmaa, H., Asokan, N., Niemi, V.: Secure vickrey auctions without threshold trust. In: Blaze, M. (ed.) FC 2002. LNCS, vol. 2357, pp. 87–101. Springer, Heidelberg (2003)

19. MacWilliams, F., Sloane, N.: The theory of error-correcting codes. North-Holland Amsterdam (1978)
20. Naor, M., Pinkas, B., Sumner, R.: Privacy preserving auctions and mechanism design. In: ACM Conference on Electronic Commerce, pp. 129–139 (1999)
21. Nguyen, K.Q., Traoré, J.: An online public auction protocol protecting bidder privacy. In: Clark, A., Boyd, C., Dawson, E.P. (eds.) ACISP 2000. LNCS, vol. 1841, pp. 427–442. Springer, Heidelberg (2000)
22. Nishide, T., Ohta, K.: Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 343–360. Springer, Heidelberg (2007)
23. Nojoumian, M., Stinson, D.R.: Unconditionally secure first-price auction protocols using a multicomponent commitment scheme. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 266–280. Springer, Heidelberg (2010)
24. Nojoumian, M., Stinson, D.R.: On dealer-free dynamic threshold schemes. Advances in Mathematics of Communications, AMC 7(1), 39–56 (2013)
25. Parkes, D.C., Rabin, M.O., Shieber, S.M., Thorpe, C.: Practical secrecy-preserving, verifiably correct and trustworthy auctions. Electronic Commerce Research and Applications 7(3), 294–312 (2008)
26. Peng, K., Boyd, C., Dawson, E.: Optimization of electronic first-bid sealed-bid auction based on homomorphic secret sharing. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 84–98. Springer, Heidelberg (2005)
27. Peng, K., Boyd, C., Dawson, E., Viswanathan, K.: Robust, privacy protecting and publicly verifiable sealed-bid auction. In: Deng, R.H., Qing, S., Bao, F., Zhou, J. (eds.) ICICS 2002. LNCS, vol. 2513, pp. 147–159. Springer, Heidelberg (2002)
28. Peng, K., Boyd, C., Dawson, E., Viswanathan, K.: Five sealed-bid auction models. In: Australasian Information Security Workshop Conference, pp. 77–86. Australian Computer Society (2003)
29. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. In: 21th Annual ACM Symposium on Theory of Computing, STOC, pp. 73–85 (1989)
30. Sako, K.: An auction protocol which hides bids of losers. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 422–432. Springer, Heidelberg (2000)
31. Shamir, A.: How to share a secret. Comm. of the ACM 22(11), 612–613 (1979)
32. Stinson, D.R., Wei, R.: Unconditionally secure proactive secret sharing scheme with combinatorial structures. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 200–214. Springer, Heidelberg (2000)
33. Suzuki, K., Yokoo, M.: Secure combinatorial auctions by dynamic programming with polynomial secret sharing. In: Blaze, M. (ed.) FC 2002. LNCS, vol. 2357, pp. 44–56. Springer, Heidelberg (2003)
34. Suzuki, K., Yokoo, M.: Secure multi-attribute procurement auction. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 306–317. Springer, Heidelberg (2006)
35. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. Journal of Finance 16(1), 8–37 (1961)
36. Viswanathan, K., Boyd, C., Dawson, E.: A three phased schema for sealed bid auction system design. In: Clark, A., Boyd, C., Dawson, E.P. (eds.) ACISP 2000. LNCS, vol. 1841, pp. 412–426. Springer, Heidelberg (2000)

37. Watanabe, Y., Imai, H.: Reducing the round complexity of a sealed-bid auction protocol with an off-line ttp. In: ACM Conference on Computer and Communications Security, CCS, pp. 80–86 (2000)
38. Yao, A.C.-C.: Protocols for secure computations. In: 23rd IEEE Annual Symposium on Foundations of Computer Science, FOCS, pp. 160–164 (1982)
39. Yokoo, M., Suzuki, K.: Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In: 1st International Joint Conference on AAMAS, pp. 112–119. ACM (2002)