Practical Receipt-Free Sealed-Bid Auction in the Coercive Environment

Jaydeep Howlader[1], Sanjit Kumar Roy[2], and Ashis Kumar Mal[3]

[1] National Institute of Technology, Durgapur, India
jaydeep.howlader@it.nitdgp.ac.in
[2] sanjit_it@yahoo.co.in
[3] ashis.mal@ece.nitdgp.ac.in

Abstract. Sealed-Bid auction is an efficient and rational method to establish the price in open market. However sealed-bid auctions are subject to bid-rigging attack. Receipt-free mechanisms were proposed to prevent bid-rigging. The prior receipt-free mechanisms are based on two assumptions; firstly, existence of untappable channel between bidders and auction authorities. Secondly, mechanisms assume the authorities to be honest (not colluding). Moreover the bandwidth required to communicate the receipt-free bids is huge. This paper presents a sealed-bid auction mechanism to resist bid-rigging. The proposed method does not assume untappable channel nor consider the authorities to be necessarily honest. The proposed mechanism also manages the bandwidth efficiently, and improves the performance of the system.

1 Introduction

Sealed-bid is a form of auction mechanism where bids are submitted in sealed-envelop. The bids are remained sealed until the schedule time of *opening*. No bids are accepted after the schedule time of *opening*. During *opening* the sealed-bids are opened and the winning price and/or winner(s) are determined. It is rather delicate to implement a sealed-bid auction in the electronic media as there are various essential security requirements to be realized. Moreover, the adversarial behavior of the entities (*insider* or *outsider*) may lead to the failure of a naively implemented system. Unlike the *outsiders'* threat, the adversarial behavior of the *insiders* are often difficult to counter. For example:

- Auctioneer (*insider*) opens the bid prior to the schedule *opening* and conveys the bid-values to the adversary [14, 4]. Thus fails to meet the *confidentiality of bid* property.
- Auctioneer allows certain bidder(s) to withdraw or submit unlawfully. Thus fails to meet the *fairness* property.
- Auctioneer deliberately suppresses some of the valid bids to make a certain bidder to be the winner. Thus fails to meet the *correctness* property.
- Auctioneer discloses all the bidding prices and the identity of the corresponding bidders after the *opening*. Thus fails to meet the *privacy of the bidder* [29] property.

- Coercer (*insider* entity) used to corrupt the authorities to retrieve critical information which may yield to bid-rigging [18]. Thus fails to meet the *uncoercibility* property.

During the last couple of decades sealed-bid auction mechanisms were studied and analyzed in various literatures. In spite of satisfying various security requirements (confidentiality, privacy, fairness, correctness etc.), sealed-bid auction mechanisms are subject to bid-rigging attack. Bid-rigging is a form of coercing where the powerful adversary (e.g. mafia) commands the other bidders to bid as per his choice so that he could win the auction by bidding unreasonably low value. Though the bids are submitted securely, coercer used to enforce the bidders to disclose all the private parameters (e.g. secret randomness, keys etc.) correspond to their secret bids. Thus coercer verifies whether the bidders obey his command. The coercer may corrupt some of the authorities and retrieves vital information that would indulge coercing.

1.1 Related work

There have been substantive research works on sealed-bid auction. Franklin & Reiter [14] first proposed a protocol for secure electronic auction. Kikuchi *et.al.* [16] proposed the multi-round auction protocol for tie-breaking. Naor *et.al.* [20] proposed a two-server auction mechanism that protected the privacy of the bidders. In the sequel we include the recent works as [27, 9, 6, 3]. However, those mechanisms have no protection to reveal the private inputs if the bidder is willing to do so. In spite of satisfying variety of security requirements, the prior mechanisms are unable to provide bid-rigging.
Abe & Suzuki [18] first introduced the receipt-free mechanism to counter bid-rigging problem. The mechanism was based on *threshold encryption* [1], with n number of auctioneers. Chen *et al.* [28] argued that Abe & Suzuki's mechanism [18] could not provide receipt-freeness to the winning bidder. Moreover, the mechanism failed to provide receipt-freeness in the presence of colluding auctioneer(s). Chen *et al.* proposed another receipt-free auction mechanism [28]. In their mechanism, seller along with the bidder jointly constructed the receipt-free bid. They argued that, seller would not be colluded due to *benefit collision*. Her *et.al.* countered their argument and showed that seller could also be colluded when she tried to make a special bidder to be the winner. Her *et al.* further proposed another receipt-free auction mechanism [29] based on anonymous channel and pseudo ID. The mechanism required prior bidders' registration. Nevertheless, their mechanism failed to provide receipt-freeness if the registrar was dishonest. Huang *et al.* [30]proposed some improvement of Abe & Suzuki's mechanism [18] while reducing the bandwidth of bids, but could not overcome the problem related to dishonest auctioneer(s). Later on Howlader *et al.* [10] attempted another receipt-free mechanism based on multi-party computation. However the mechanism failed to provide receipt-freeness as the bidders' verification process carried the receipt of the bid.

Constraints & Assumptions	Abe & Suzuki [18]	Chen *et al.* [28]	Her *et al.* [29]	Huang *et al.* [30]	Howlader *et al* [10]	Gao *et al.* [8]
Untappable channel	one way	both way	one way	one way	one way	not specified
Anonymity	×	×	$\surd$	×		×
Honest Authority	All honest auctioneer	honest seller	honest registrar	all honest auctioneer	at least one honest sealer	honest auctioneer
Bandwidth	$O(l \times n)$	$O(l)$	$O(c)$	$O(log\ l \times n)$	$O(l)$	$O(c)$

Table 1: The physical constrains and assumptions made in various sealed-bid auction mechanisms. l denotes the length of the price list, n denotes the number of auctioneers and c denotes constant

Some impractical assumptions: The above mechanisms are based on the two assumptions:
Firstly, the availability of untappable channel [4] between bidders and authorities (auctioneers, seller, sealer etc.). However, untappable channel is often impractical and difficult to deploy. However, some techniques based on deniable encryption [21, 12] ware proposed in [24, 13] to relax the untappable channel. The notion of deniability allows the bidders to plausibly evade the coercer. However, those techniques fail in the presence of colluding authorities [11]. Later on Howleder *et. al.* introduced *'Coercing Resistant Mix (CRM)'* [11] which integrated deniability with anonymity to transients the physical requirement of untappable channel.
Secondly, the prior receipt-free mechanisms consider the authorities to be honest. More specifically, the authorities not only execute the protocol honestly, but also avoid any such conspiracy that may leak certain information to the coercer.

1.2 Our Contribution

We withdraw the untappable channel, henceforth coercer can intercept the public transcripts at any extend. Furthermore, we consider a broader notion of coerciveness rather than only receipt-freeness. Coercer may collude some of the authorities who execute the protocol correctly but reveal certain information to the coercer in order to indulge coercing. We replace the untappable channel with CRM [11]. CRM allows the adversary to intercept the public transcripts, but provides the bidders to formulate *'fake bids'* such that, adversary could not able to distinguish between the *fakes* and the *trues*. On the other hand, untraceable delivery of messages restricts the recipient (authority) to link 'who-bids-what'. Though the recipient recives decrypted messages, but unable to determine 'who-bids-what'.

Based on the cryptographic techniques, the receipt-free auction mechanisms are categorized in two classes. The mechanisms [18, 30] are based on threshold secret sharing which outputs committed transcripts. However, those mechanisms fail to provide receipt-freeness if any one of the authority reveals his share. Whereas the mechanisms [28, 29] are based on designated-verifiability of

[4] A channel that provides perfect security in an information-theoretic sense. Even encryption does not provide an untappable channel

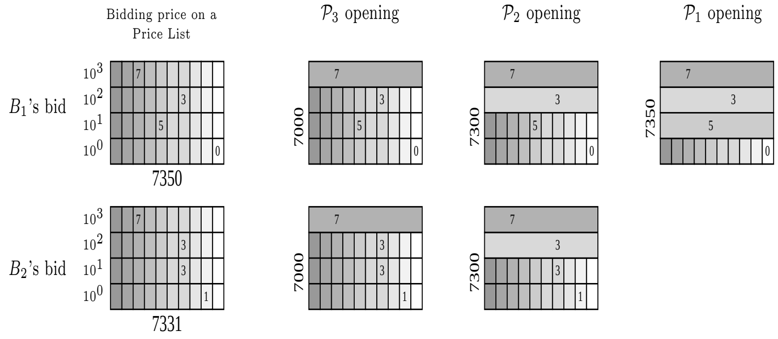

Fig. 1: Opening of two bid-vectors B_1 : 7350 and B_2 : 7331. During the opening of $\mathcal{P}_1$ the bid-vector B_1 is extracted while B_2 is excluded as $P_{1,5}$ appears before $P_{1,3}$.

re-encryption proof [2], where bidder and authority (either seller, auctioneer) collaboratively form the receipt-free bids. Nevertheless, those schemes also fail if the entities are not trustworthy.

The proposed mechanism is based on secure multi-party computation [5, 25]. The sealing operation is done with respect to a private key which is distributed among a set of qualifying sealers. A quorum of qualifying sealers performs the sealing operation form the sealed-bids. Unlike the prior mechanisms the proposed scheme guarantees receipt-freeness even at least one of the authority remain honest (not colluded).

2 Preliminaries

Three main building blocks are used in the proposed receipt-free mechanism. They are Deniable Encryption, Coercer Resistant MIX and Distributed Key Generation.

A **Plan-Ahead Deniable Encryption (PDE)** [21, 12] outputs the cipher c_d such that, the encryption of the fake and the true messages look alike. The PDE consists of three algorithms $PDE(Enc, Dec, \varphi)$. The encryption (Enc) is defined as $Enc^{-}(m_t, pk, r_t)$, where m_t is the true message, pk is the public key and r_t is the true randomness, and outputs a cipher c. However, Enc produces deniable cipher c_d when executed with another parameter called fake message m_f, as $Enc^{m_f}(m_t, pk, r_t)$. PDE allows the sender to evade coercion by producing m_f instade of m_t. The decryption (Dec) is defined as $Dec(c \text{ or } c_d, sk)$, where sk is the private key and outputs the plaintext m_t with negligible decryption error. The faking algorithm(φ) is defined as $\varphi(c_d, m_t, m_f)$ and outputs the fake randomness r_f such that $Enc^{-}(m_t, pk, r_t)$ and $Enc^{-}(m_f, pk, r_f)$ look alike.

MIX (MIX-cascade) is a system consists of a finite number of nodes and provides anonymous communication [17, 19, 15]. MIX takes a list of ciphertexts as input and outputs a random permutation of the plaintexts. Every node performs a cryptographic transformation and a random permutation, and forwards the list to the next node. We denote MIX operation as $MIX(Enc_{pk}[m_1, \ldots, m_N]) \rightarrow \prod[m_1, \ldots, m_N]$ where pk is the public key of the MIX and $\prod$ denotes a random permutation of the list. Unlike the general MIXes those take non-probabilistic ciphers [25, 26][5] as input, CRM takes deniable ciphers as input. Deniability allows the sender to plausibly deny the true message while anonymity restricts the dishonest recipients to retrace the senders of individual messages.

Distributed Key Generation (DKG) allows a set of n entities to generate jointly a pair of public-private key according to the distribution defined by the underlying cryptosystem. The public key is output in the clear, the private key is secretly shared among the n entities via a threshold encryption scheme. A robust and efficient DKG protocol is proposed by Gennaro et.al. [23] to share the secret x amongst a set of qualifying entities and the makes $y = g^x$ public. The protocol is able to identify the malicious entities and computes the public-private values with the inputs of the qualifying entities. DKG is denoted as $DKG_P(s_1, \ldots, s_n) \rightarrow (h, x, \mathcal{P})$, where P is the set of n entities, s_i is the random secret initiated by the entity $P_i \in P$, $h = g^x$ is the public value, $x = f_{s_i \in \mathcal{P}}(s_i)$ is the secret shared amongst the entities $P_i \in \mathcal{P}$ and $\mathcal{P}$ is the set of qualifying entities.

3 Receipt-free Sealed-Bid Auction

The receipt-freeness is proposed to prevent bid-rigging in sealed-bid auction. Following are the entities of the proposed receipt-free auction:

3.1 Entities

- There is a finite set of bidders denoted as $B = \{B_1, B_2, \ldots, B_m\}$.
- There is a finite set of sealers denoted as $S = \{S_1, S_2, \ldots, S_k\}$. Sealer is an authority who executes sealing operation and forms the receipt-free bid.
- There is a single auctioneer. The auctioneer is responsible to open the bids (with the cooperation of sealers) and determines the winning price and winner.
- Coercer is an adversary who indulges bid-rigging. Coercer is able to impel the bidders to reveal all their private data (keys and randomnesses). Furthermore the coercer is allowed to intercept the public transcripts and also corrupts some of the sealers to retrieve critical information that may yield coercing.
- We use a **Bulletin Board** ($\mathcal{BB}$). This is a publicly accessible memory with read and appendive-write access.
- We integrate CRM in place of untappable channel

[5] probabilistic encryption uses randomness in encryption so that, when encrypting the same message several times it will, in general yield different ciphertexts

Algorithm 1: Bidder B_i bidding operation

```
begin
    for $k = d-1$ to $0$ do
        for $j = 9$ to $0$ do
            $B_i$ randomly selects $r_{i,(k,j)}, \hat{r}_{i,(k,j)} \in_R \mathbb{Z}_p^*$ and computes $\left(X_{i,(k,j)}, Y_{i,(k,j)}\right)$
            $X_{i,(k,j)} = g^{r_{i,(k,j)}}$
            $Y_{i,(k,j)} = \begin{cases} (h_A.h_S)^{r_{i,(k,j)}}.G_{i,(k,j)} & \text{if } j = \delta_k \\ (h_A.h_S)^{r_{i,(k,j)}} & \text{otherwise} \end{cases}$
        // where $G_{i,(k,j)} = \hat{r}_{i,(k,j)} G_i^{r_{i,(k,j)}}$ represents the Yes mark, $G_i = g_y^{x_{B_i}}$
    $B_i$ outputs the encrypted bid-vector $\langle \mathcal{X}_i, \mathcal{Y}_i \rangle$ corresponds to the price list $\mathcal{P}$
end
```

3.2 System Setting

Let p, q be large primes such that q divides $p-1$, G_q be the unique subgroup of $\mathbb{Z}_p^*$ of order q, and $g \in \mathbb{Z}_p^*$ is an element of order q. Following we define the keys of different entities. The operations are closure to the multiplicative group $\mathbb{Z}_p^*$.
• Bidder B_i's private key be $x_{B_i} \in \mathbb{Z}_p^*$ and public key be $h_{B_i} = g^{x_{B_i}}$.
• Auctioneer's private key be $x_A \in \mathbb{Z}_p^*$ and public key be $h_A = g^{x_A}$.
• The sealers execute the Distributed Key Generation (DKG) protocol [23, 22] that outputs a set of qualifying sealers denoted as $QUAL$ (of k sealers) with the public key h_S. Each member $S_i \in QUAL$ has his private key as x_i such that any quorum of $t > k/2$ sealers denoted as $QRM \subseteq QUAL$ are able to seal the bidders' encrypted bid-vectors. Without loss of generality, we assume that $QRM = \{S_1, S_2, ..., S_t\}$. Sealer $S_i \in QRM$ configures his sealing key as $x_{S_i} = f_i(0)$ where $f_i(x) = \lambda_{ij} x_i$ is a polynomial of degree t. λ_{ij} [6] is the Lagrange interpolation coefficient for the sealer S_i.
•After configuring the QRM each sealer $S_i \in QRM$ publishes his public key for sealing as $h_{S_i} = g^{x_{S_i}}$. We denote $h_{S/S_1,S_2,...,S_r} = h_S(h_{S_1} h_{S_2} \dots h_{S_r})^{-1}$. Intuitively $h_{S/S_1,...,S_t} = 1$
• $g_y \in \mathbb{Z}_p^*$ be an element of order q indicates the *YES Mark*.
• Let the maximum estimated price of the item is lesser the 10^d. Auctioneer publishes the price list $\mathcal{P}$ consisting of d ordered vectors. We denote $\mathcal{P} := \mathcal{P}_{d-1}, \mathcal{P}_{d-2}, \dots, \mathcal{P}_0$ where every $\mathcal{P}_i$ consists of 10 elements and denoted as $\mathcal{P}_i := P_{i9}, P_{i8}, \dots, P_{i0}$. The element P_{ij} represents the value $j \times 10^i$. Thus the decimal value of $\delta_{d-1}\delta_{d-2}\dots\delta_0$ has an equivalent representation as $\sum_{i=0}^{d-1} P_{i\delta_i}$. Fig 1 describes the bid-vector representation of the decimal value (bid value).

4 Receipt-free sealed-bid auction mechanism

The receipt-free sealed-bid auction mechanism is consisting of four phases: *bidding*, *sealing*,*opening* and *trading*.

[6] Lagrange interpolation coefficient for the i^{th} sealer is $\lambda_{ij} = \prod_{\substack{i \neq j \\ 1 \leq j \leq t}} \frac{x-j}{i-j}$

Algorithm 2: Sealing operation

```
begin
    if ($S_{l=1}$ is the first Sealer $\in QRM$) then
        $S_l$ receives $\langle \mathcal{X}_i, \mathcal{Y}_i \rangle$ and computes $\langle \mathcal{X}_{S_l i}, \mathcal{Y}_{S_l i} \rangle$ as follows
        for $k = d - 1$ to $0$ do
            for $j = 9$ to $0$ do
                $S_l$ randomly selects $r_{S_l i,(k,j)}, \hat{r}_{S_l i,(k,j)} \in_R \mathbb{Z}_p$ and computes
                $$\begin{aligned} X_{S_l i,(k,j)} &= g^{r_{S_l i,(k,j)}} X_{i,(k,j)} \\ Y_{S_l i,(k,j)} &= \hat{r}_{S_l i,(k,j)} . h_A^{r_{S_l i,(k,j)}} . h_{S/S_l}^{r_{S_l i,(k,j)}} . (X_{i,(k,d)})^{-x_{S_l}} . Y_{i,(k,d)} \end{aligned} \quad (1)$$
        // We denote $G_{\square} = \begin{cases} G_{i,(k,j)} & \text{if } B_i \text{ has marked } P_{(k,j)} \text{ with YES} \\ 1 & \text{otherwise} \end{cases}$
        $S_{l=1}$ forwards the partially sealed bid-vector $\langle \mathcal{X}_{S_l i}, \mathcal{Y}_{S_l i} \rangle$ corresponds to $\langle \mathcal{X}_i, \mathcal{Y}_i \rangle$ to the next Sealer
    else if ($S_{l \neq 1}$ is the intermediate Sealer $\in QRM$) then
        $S_l$ receives the partially sealed bid-vector from $S_{l-1}$ as $\langle \mathcal{X}_{S_{l-1} i}, \mathcal{Y}_{S_{l-1} i} \rangle$ and computes $\langle \mathcal{X}_{S_l i}, \mathcal{Y}_{S_l i} \rangle$ as follows
        for $k = d - 1$ to $0$ do
            for $j = 9$ to $0$ do
                $$\begin{aligned} X_{S_l i,(k,j)} &= g^{r_{S_l i,(k,j)}} . X_{S_{l-1} i,(k,j)} \\ Y_{S_l i,(k,j)} &= \hat{r}_{S_l i,(k,j)} . h_A^{r_{S_l i,(k,j)}} . (h_{S/S_1,\ldots,S_l})^{r_{S_l i,(k,j)}} . \\ &\quad (X_{S_{l-1} i,(k,d)})^{-x_{S_l}} . Y_{S_{l-1} i,(k,d)} \end{aligned} \quad (2)$$
        if ($S_{l=t}$ is the last sealer $\in QRM$) then
            $S_{l=t}$ publishes the sealed bid-vector $\langle \mathcal{X}_{S_t i}, \mathcal{Y}_{S_t i} \rangle$ on the $\mathcal{BB}$
        else
            $S_l$ forwards the partially sealed bid-vector to the next sealer
end
```

Bidding: Every bidder $B_i \in B$ determines his bidding price and constructs the encrypted bid-vector as follows:

- Let $\delta_{d-1}\delta_{d-2}\ldots\delta_0$ $(0 \leq \delta_i \leq 9)$ be the decimal representation of the bidding price. B_i executes Algorithm 1 to output the encrypted bid-vector $\langle \mathcal{X}_i, \mathcal{Y}_i \rangle$.
- B_i marks the price indices $P_{k\delta_k}$ $(0 \leq k \leq d-1)$ with *Yes* while encrypting the price list $\mathcal{P}$.

B_i constructs a fake encrypted bid-vector as $\langle \bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i \rangle$ and forwards the deniable cipher $Enc^{\langle \bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i \rangle}(\langle \mathcal{X}_i, \mathcal{Y}_i \rangle, pk_{CRM}, r_t)$ to the CRM. CRM accumulates a batch of deniable ciphers and anonymously delivers the batch to the $QUAL$.

Sealing: A quorum of sealers (denoted as $QRM \subset QUAL$), possessing the public key as h_S, performs the sealing operation. The encrypted bid-vectors are processed by at least $t > k/2$ sealers from the QRM. Every sealer $S_l \in QRM$ executes the Algorithm 2 and outputs the partially sealed bid-vector. During *sealing*, every sealer S_l engraves his secret randomness, $r_{S_l i,(k,j)}$ & $\hat{r}_{S_l i,(k,j)}$ and nullifies his key component, h_{S_l} (in Algorithm 2, equation 1 & 2) from the partially sealed bid-vector. After t sealing operation sealer S_t publishes the sealed-bid on the $\mathcal{BB}$.

Bid Verification (BV): The inherent property of receipt-freeness is the inability to prove to any one how a bidder has bid. However, receipt-freeness allows the bidder to verify the correctness of the sealing operation. Algorithm 3 describes the BV mechanism. The BV does not reveal the secret value i.e, even the coercer observes the process of BV, bidder can execute the BV correctly without revealing any partial information related to his secret. BV is done with respect to the cumulative response $\mathcal{R}_{S_\mathbf{t}ik}$ computed by every sealers.

Opening: At the schedule *Opening*, bids are opened. Bids are opened in decreasing order (starting from the highest price). We define two subprocesses: Evaluating *Yes Mark* (EYM) for the price vector $\mathcal{P}_k$ and Extracting Bids having *Yes Mark* (EBY) on price index $P_{k,j}$.

- EYM: Auctioneer and sealers jointly execute the process. EYM takes input a price vector $\mathcal{P}_k$ and output the highest price index $P_{k,j}$ that contains some *Yes Marks*. Algorithm 4 describes the process.
- EBY: After EYM outputs an index $P_{k,j}$, EBY extracts and outputs a list of bids (sealed-bids) that contains the sealed-bids having *Yes Mark*s on the $P_{k,j}$ index.

The *opening* phase is initiated with the construction of the list L containing all the sealed bids followed by invoking the subporcess $EYM(L, \mathcal{P}_{d-1})$. EYM will output the price index $P_{d-1,w_{d-1}}$ and the list L_{d-1} containing those sealed bids which possess *Yes Mark* on the index $P_{d-1,w_{d-1}}$. Auctioneer sets the winning price as $w = w_{d-1}xxx$. Auctioneer subsequently iterates the subprocess $EYM(L_k, \mathcal{P}_k)$ (for $k = d-2, \ldots 0$) and finally the winning price $w = w_{d-1} \ldots w_0$ and the list of winning bids L_0 is determined.

Trading: The auction mechanism determines the winning bids, but not the winner. The winning bidder claims his winning and executes a zero-knowledge (ZK) protocol with the auctioneer to substantiate his winning. Let B_i be the winning bidder and $w = w_{d-1} \ldots w_0$ be the winning price. Bidder B_i proves
G_i and h_{B_i} have common exponent over g_y and g respectively.
For $k = d-1, \ldots 0$, B_i discloses all $\hat{r}_{i,(k,w_k)}$ and proves that $X_{i,(k,w_k)}$ and $G_{i,(k.w_k)} \cdot (\hat{r}_{i,(k,w_k)})^{-1}$ have common exponents over g and G_i respectively.
The details of the ZK protocol is presented in the Appendix.

5 Security Analysis

In this section we present the security properties of the proposed scheme:

Receipt-Freeness: If A is an auction protocol and simulated as

$$A \triangleq Bid(\forall i B_i, b_i, r_{B_i}) | Seal(\forall t S_t, r_{S_t}, \hat{r}_{S_t}) | out(sb_i) |$$
$$Rev(B_c, r_{B_c}) | !Rev(\exists h S_h, r_{S_h}, \hat{r}_{B_h}) | Rev(\forall t S_t, r_{S_t}, \hat{r}_{S_t})$$

Algorithm 3: Bid Verification

```
begin
    for $\forall S_l \in QRM$ do
        if $(S_{l=1}$ is the first Sealer $\in QRM)$ then
            $S_{l=1}$ computes the response-vector $\mathcal{R}_{S_l i,k}$ as follows
            for $k = d-1$ to $0$ do
                $\mathcal{R}_{S_l i,k} = \Big(\prod_{j=0}^{9} \hat{r}_{S_l i,(k,j)}\Big)$
            $S_{l=1}$ appends the response-vector with the partially sealed bid-vector as
            $\langle \mathcal{X}_{S_l i,(k,-)}, \mathcal{Y}_{S_l i,(k,-)} \rangle \mathcal{R}_{S_l i,k}$ (for $0 \le k \le d-1$) and forwards.
        else if $(S_{l \neq 1}$ is the intermediate Sealer $\in QRM$ then
            $S_l$ receives $\langle \mathcal{X}_{S_{l-1} i,(k,-)}, \mathcal{Y}_{S_{l-1} i,(k,-)} \rangle \mathcal{R}_{S_{l-1} i,k}$ and computes his
            response-vector $\mathcal{R}_{S_l ik}$ as follows
            for $k = d-1$ to $0$ do
                $\mathcal{R}_{S_l i,k} = \Big(\prod_{j=0}^{9} \hat{r}_{S_l i,(k,j)}\Big).\mathcal{R}_{S_{l-1} i,k} = \Big(\prod_{j=0}^{9}\prod_{t=1}^{l} \hat{r}_{S_t i,(k,j)}\Big)$
            $S_l$ overwrites the preceding response-vector $\mathcal{R}_{S_{l-1} i,k}$ with his response-vector
            as $\langle \mathcal{X}_{S_l i,(k,-)}, \mathcal{Y}_{S_l i,(k,-)} \rangle \mathcal{R}_{S_l i,k}$ (for $0 \le k \le d-1$) and forwards.
        if $S_{l=t}$ is the final sealer $\in QRM$ then
            $S_t$ publishes $\langle \mathcal{X}_{S_t i,(k,-)}, \mathcal{Y}_{S_t i,(k,-)} \rangle \mathcal{R}_{S_t i,k}$ on $\mathcal{BB}$
    // After all sealers comput their responses, Auctioneer blindly signs the response
       as follows
    for $\forall i$ sealed-bid vectors $\langle \mathcal{X}_{S_t i,(k,-)}, \mathcal{Y}_{S_t i,(k,-)} \rangle \mathcal{R}_{S_t i,k}$ Auctioneer computes do
        for $k = d-1$ to $0$ do
            $\mathbb{X}_{i,k} = \left(\prod_{j=0}^{9} X_{S_t i,(k,j)}\right)^{x_A} = h_A^{\sum_{j=0}^{9}\left(r_{i,(k,j)} + \sum_{l=1}^{t} r_{S_l i,(k,j)}\right)}$
        Auctioneer appends the blind signature with the sealed bid-vectors as
        $\langle \mathcal{X}_{S_t i,(k,-)}, \mathcal{Y}_{S_t i,(k,-)} \rangle \mathcal{R}_{S_t i,k}, \mathbb{X}_{i,k}$ (for $0 \le k \le d-1$) and publishes on $\mathcal{BB}$.
    // After Auctioneer publishes the blind signatures, Bidder $B_i$ verifies his sealed
       bid as follows
    for $l = 1$ to $m$ do
        Bidder $B_i$ set $VEFY = TRUE$
        for $k = d-1$ to $0$ do
            if $\Big(\prod_{j=0}^{9} Y_{S_t l(k,j)}! = \mathcal{R}_{S_t lk}.\mathbb{X}_{lk}.\prod_{j=0}^{9} G_{i,(k,j)}\Big)$ then
                $VEFY = VEFY \cap FALSE$
        if $(VEFY == TRUE)$ then
            $B_i$ verifies and RETURN
    if $VEFY == FALSE$ then
        $B_i$ raises a complain
end
```

```
**ProcS**$(L, S_l, P_{k,j})$
$S_l$ computes $V_{S_l(k,j)} = \prod_{\langle \mathcal{X}_i, \mathcal{Y}_i \rangle \in L} \hat{r}_{S_l i,(k,j)}$
Sealer $S_l$ outputs $V_{S_l(k,j)}$ on the $\mathcal{BB}$
```

```
**ProcA**$(L, P_{k,j})$
Auctioneer computes $\mathbb{V}_{(k,j)} = \prod_{1}^{t} V_{S_l(k,j)}$
    $= \left( \prod_{i=1}^{m} \prod_{1}^{t} \hat{r}_{S_l i,(k,j)} \right).$
Furthermore auctioneer computes
$\mathbb{Y}_{(k,j)} = \prod_{b_i \in L} Y_{S_t i,(k,j)} X_{S_t i,(k,j)}^{-x_A}$
    $= \left( \prod_{b_i \in L} \prod_{l=1}^{t} \hat{r}_{S_l i,(k,j)} \right)$
    $= g^{\sum_{b_i \in L} \sum_{l=1}^{t} r_{S_l i,(k,j)}} \cdot \prod_{b_i \in L} G_{\square i,(k,j)}$
Auctioneer evaluates and outputs
$\mathbb{G}_{(k,j)} = \mathbb{Y}_{(k,j)} . \mathbb{V}_{(k,j)}^{-1}$
Auctioneer outputs $\mathbb{V}_{(k,j)}$,$\mathbb{Y}_{(k,j)}$ on $\mathcal{BB}$
```

```
**ProcSwap**$(L = \{b_1, b_2\}, L_{void} = \{v_1, v_2\}$
// $L$: list having Yes mark
// $L_{void}$: void list
Construct $\bar{L} = \{b_1, v_2\}$ & $\bar{\bar{L}} = \{v_1, b_2\}$
for $l = 1$ to $t$ do
    $ProcS(\bar{L}, S_l, P_{k,j})$
    $ProcS(\bar{\bar{L}}, S_l, P_{k,j})$
end
if $ProcA(\bar{L}, P_{k,j}) \neq 1$ then
    $b_1$ has *Yes Mark* and included
    in the output list
end
if $ProcA(\bar{\bar{L}}, P_{k,j}) \neq 1$ then
    $b_2$ has *Yes Mark* and included
    in the output list
end
```

Algorithm 4: EYM$(L, \mathcal{P}_k)$

```
Input: $L = \{b_i\}$ be stack of sealed bid
for $j = 9$ to $0$ do
    // Every Sealer $S_t \in QRM$
    // executes $ProcS()$
    for $l = 1$ to $t$ do
        $ProcS(L, S_l, P_{k,j})$
    // Auctioneer executes $ProcA()$
    $ProcA(L, P_{k,j}) \rightarrow \mathbb{G}$
    if ($\mathbb{G} \neq 1$) then
        $EBY(L, P_{k,j})$
```

Algorithm 5: $EBY(L, P_{k,j})$

```
List $L$ is divided in two
halves $L1$ & $L2$
// Every sealer $S_l \in QRM$
// executes $ProcS()$ for $L1$ & $L2$
for $l = 1$ to $t$ do
    $ProcS(L1, P_{k,j})$;$ProcS(L2, P_{k,j})$;
// Auctioneer executes $ProcA()$
// for the two halves
$ProcA(L1, P_{k,j}) \rightarrow \mathbb{G}_1$
$ProcA(L2, P_{k,j}) \rightarrow \mathbb{G}_2$
if ($\mathbb{G}_1 \neq 1$) then
    if ($|L1| \geq 2$) then
        $EBY(L1, P_{k,j})$
    else
        ProcSwap$(L1, L_{void})$
if ($\mathbb{G}_2 \neq 1$) then
    if ($|L2| \geq 2$) then
        $EBY(L2, P_{k,j})$
    else
        ProcSwap$(L2, L_{void})$
```

where every bidder B_i encrypts his bid b_i with the randomness r_{B_i}, every sealer S_t seals the bids with randomness $r_{S_i}, \hat{r}_{S_i}$ and produces the sealed bids sb_i, thereafter, the coerced bidder B_c and all sealers except the honest sealer S_h reveal their secrets. The protocol still conceals the private values. We show that, adversary who may compute buy could not resolve the secret as the private values are

$$Y_{S_t c,(k,j)} . \Big(\prod_{l=1 l \neq h}^{t} \hat{r}_{S_l c,(k.j)} . h_A^{\left(r_{c,(k,j)} + \sum_{l=1 l \neq h}^{t} r_{S_l c,(k,j)} \right)} \Big)^{-1} = \hat{r}_{S_h c,(k,j)} . G_{\Box}$$

blinded with the honest bidder's randomness $(\hat{r}_{S_h c(k,j)})$. Similarly, bidder B_c could flip the *Yes-to-No* and vice-versa. The proposed mechanism ensures receipt-freeness as adversary could not distinguish between a situation where B_c reveals his true secret and the situation where he produces fake secret.

Correctness: The auction mechanism declares the winning price and keeps all the loosing bids secret. The *correctness* defines the ability to verify the outcome of the auction by any entity. Let auctioneer declared $w = w_{d-1}w_{d-2}\dots w_0$ as the winning price. Therefore during *opening* the subprocesses **ProcS()** and **ProcA()** have published all $V_{S_i(k,j)}$ and $\mathbb{V}_{(k,j)}$ & $\mathbb{Y}_{(k,j)}$ $(0 \leq k \leq d-1,\ 9 \geq j \geq w_k)$ on $\mathcal{BB}$ on the $\mathcal{BB}$. Any one who wants to verify the correctness of the auction result can examine the result with the information published on the $mathcalBB$.

Nonrepudiation: We assume that bidder bids honestly. The *opening* of bids only determines the winning price and the list of winning bids, but winner is not determined. Bidder executes the ZK protocol to substantiate his winning. However, the odd may happen, when the winning bidder does not respond. We present the mechanism to identify the winning bidder while he has not responded. Let $w = w_{d-1}w_{d-2}\dots w_0$ be the winning price and L_0 be the list of sealed-bids extracted as the winning bid(s). Let $\langle \mathcal{X}_{S_t i}, \mathcal{Y}_{S_t i} \rangle \in L_0$ be an winning bid. In the *opening* phase, procedure **ProcSwap()** computes the *Yes Mark* on every P_{k,w_k} of the winning bid. Let $\mathcal{G} = \{G_{i(k,w_k)} \mid 0 \leq k \leq d-1\}$ be the set of *Yes Mark*s computed by **ProcSwap()** during *opening*. Now auctioneer has to identify the bidder(s) who had bid with the above set of *Yes Mark*s. Auctioneer initiates the following:

- Auctioneer asks all sealer $S_l \in QRM$ to publish the initial encrypted price-vectors on $\mathcal{BB}$. Thus all $\langle \mathcal{X}_i, \mathcal{Y}_i \rangle$ (for $i = 1, 2, \dots m$) appears on the $\mathcal{BB}$.
- Auctioneer asks the bidders to substantiate their encrypted bids for every P_{k,w_k} indices . That is, all the losing bidder B_i will show that:
  1. he knows the discrete logarithm of $X_{i(k,w_k)}$ (say $r_{i(k,w_k)}$) and
  2. shows that $Y_{i(k,w_k)} = (h_S h_A)^{r_{i(k,w_k)}}$.

  However, the winning bidder B_w will fail to establish the second as he had computed $Y_{w(k,w_k)} = (h_S h_A)^{r_{w(k,w_k)}}.G_{w(k,w_k)}$.

6 Performance

The proposed scheme improves the performance by reducing the bandwidth of the receipt-free bids. The existing receipt-free auction mechanisms e.g. [18,

No. of Rounds	Abe & Suzuki [18]	Huang *et al.* [30]	proposed mechanism
During Bidding	nmL	$nm(log\ L)$	$m(log_{10}\ L)$
During Opening	$t \geq n/2$	$t \geq n/2$	worst case $10d$ and dm

No. of rounds)	Chen *et al.* [28]	Her *et al.* [29]
During Bidding	at least mnL	at least mn
During Opening	at most L	at most L

Table 2: Number of message exchanges in various auction mechanisms

	Howlader et. al. [10]				proposed mechanism			
key size (bits)	Bidding time in sec.		Sealing time in sec.		Bidding time in sec.		Sealing time in sec.	
	price list length		price list length		price list length		price list length	
	5,000	10,000	5,000	10,000	5,000	10,000	5,000	10,000
512	14	29	37	75	0.22	0.22	0.48	0.48
1024	98	195	248	495	1.48	1.48	2.76	2.76
1536	308	615	766	1515	4.63	8.45	5.69	10.24

Table 3: Time latency for *Bidding* and *Sealing* operation

28, 10] require huge bandwidth to communicate the encrypted bids. Table 1 presents the bandwidth requirement for various auction mechanisms. In this section we analyze the bandwidth requirement, communication overhead and computational complexity of the proposed mechanism. Let L, n and m represent the number of bidding price, number of auctioneer/sealer and number of bidder respectively.

We represent the price list as a d-tuples of constant length ordered-vectors. A price list of d vectors is capable to represent the value up to 10^d. Reduction in the size of price list decreases the bandwidth requirement and computational overhead. We estimate bandwidth of every receipt-free bid is $O(log_{10}\ L)$.

Moreover, The proposed mechanism defines less number of message exchange between the entities. Table 2 presents the number of message exchanges required to execute the *proofs & verification*. We also present the average time latency of bidding and sealing operation with varying key size and number of bidding price. Table 3 shows the comparison of time latency between to mechanisms.

7 Conclusion

The proposed auction scheme attempts to solve two existing problems; firstly, it provides receipt-freeness without any untappable channel and secondly, it ensures uncoerciveness even in the presence of colluding authorities. The mechanism guarantees uncoerciveness even all the sealer except one are dishonest. No prior registration of bidder is required. So any one who possesses the required key may participate in the auction. Bidders are not necessarily be present during *opening* i.e. ensures 'bid-and-go' concept. The proposed mechanism improves the performance and efficiency by reducing the bandwidth and communication round.

References

1. Shamir A. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
2. Lee B. and Kim K. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In *ICISC'02*, LNCS2587, pages 389–406. Springer-Verlag, 2002.
3. Mihály Bárász, Péter Ligeti, László Mérai, and Daniel A. Nagy. Anonymous sealed bid auction protocol based on a variant of the dining cryptographers' protocol. *Periodica Mathematica Hungarica*, 65(2):167–176, 2012.
4. Boyd C. and Mao W. *Security Issues for Electronic Auctions*. HP Laboratories technical report. Hewlett-Packard Laboratories, 2000.
5. Yao Andrew C. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164. IEEE Computer Society, 1982.
6. Wu C-C., Chang C-C., and Lin I-C. New sealed-bid electronic auction with fairness, security and efficiency. *J. Comput. Sci. Technol.*, 23(2):253–264, 2008.
7. Chaum D. and Pedersen T. P. Wallet database with observers. In *Advances in Cryptology, CRYPTO 92*, LNCS 740, pages 89–105, 1993.
8. Chongzhi Gao, Zheng an Yao, Dongqing Xie, and Baodian Wei. Electronic sealed-bid auction with incoercinility. In *Electronic Power Systems and Computers*, LNEE 99, pages 47–54. Springer-Verlag, 2011.
9. Xiong H., Qin Z., Zhang F., Yang Y., and Zhao Y. A sealed-bid electronic auction protocol based on ring signature. In *ICCCAS*, pages 480–483. IEEE, 2007.
10. Howlader J., Ghosh A., and Pal T. D. Secure receipt-free sealed-bid electronic auction. In *IC3*, CCIS 40, pages 228–239. Springer, 2009.
11. Howlader J., Kar J., and Mal A. K. Coercion resistant mix for electronic auction. In *ICISS*, LNCS 7671, pages 238–248. Springer, 2012.
12. Howlader J. and Basu S. Sender-side public key deniable encryption scheme. In *ARTCom*, pages 9–13. IEEE Computer Society, 2009.
13. Howlader J., Nair V., Basu S., and Mal A. K. Uncoercibility in e-voting and eauctioning mechanisms using deniable encryption. *IJNSA*, 3(2):97–109, 2011.
14. Franklin M. K. and Reiter M. K. The design and implementation of a secure auction service. *IEEE Transactions on Software Engineering*, 22(5):302–312, 1996.
15. Sako K. and Kilian J. Receipt-free mix-type voting scheme: a practical solution to the implementation of a voting booth. In *EUROCRYPT*, pages 393–403, 1995.
16. Hiroaki Kikuchi, Michael Hakavy, and Doug Tygar. Multi-round anonymous auction protocols. *Institute of Electronics, Information, and Communication Engineers Transactions on Information and Systems*, E82-D(4):769–777, April 1999.
17. Chaum D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.
18. Abe M. and Suzuki K. Receipt-free sealed-bid auction. In *ISC*, LNCS 2433, pages 191–199. Springer, 2002.
19. Jakobsson M. A practical mix. In *EuroCrypt*, pages 448–461, 1998.
20. Moni Noar, Benny Pinkas, and Reubew Sumner. Privacy preserving auction and mechanism design. In *ACM Conference on Electronic Commerce*, pages 129–139. ACM, 1999.
21. Canetti R., Dwork C., Naor M., and Ostrovsky R. Deniable encryption. In *CRYPTO 97*, pages 90–104, 1997.
22. Gennaro R., Jarecki S., Krawczyk H., and Rabin T. Secure distributed key generation for discrete-log based cryptosystems. In *EUROCRYPT*, LNCS 1592, pages 295–310. Springer, 1999.

23. Gennaro R., Jarecki S., Krawczyk H., and Rabin T. Secure distributed key generation for discrete-log based cryptosystems*. *Journal of Cryptology*, 20(1):51–83, 2007.

24. Zuzana Rjašková. Electronic voting schemes. Master's thesis, Department of Computer Science Faculty of Mathematics, Physics and Informatics Comenius University, Bratislava, 2002.

25. Goldwasser S. and Micali S. How to play any mental game or a completeness theorem for protocols with honest majority. In *19th annual ACM symposium on Theory of computing*, pages 365–377. ACM, 1982.

26. Goldwasser S. and Micali S. Probabilistic encryption. *Journal of Computer and Systems Sciences*, 28(2):270–299, 1984.

27. Ham W., Kim K., and Imai H. Yet another strong sealed-bid auctions. In *SCIS*, page 1116, 2003.

28. Chen X., Lee B., and Kim K. Receipt-free electronic auction schemes using homomorphic encryption. In *ICISC*, LNCS 2971, pages 259–273. Springer, 2003.

29. Her Y-S., Imamoto K., and Sakurai K. Receipt-free sealed-bid auction based on mix-net and pseudo id, 2004.

30. Huang Z., Qiu W., Guan H., and Chen K. Efficient receipt-free electronic auction protocol. In *SITIS*, pages 1023–1028. IEEE Computer Society, 2007.

Appendix

Proof of Sealing

Sealer S_l receives the partially sealed bid-vector $\langle \mathcal{X}_{S_{l-1}i}, \mathcal{Y}_{S_{l-1}i} \rangle$ from the preceding sealer S_{l-1}, selects $\hat{r}_{S_l i,(k,j)}, r_{S_l i,(k,j)} \in_R \mathbb{Z}_p$ randomly, performs the sealing operation and forwards the partially sealed bid-vector to the next sealer S_{l+1}. Fig. 2 describes the process. The sealing operation of the S_l is as follows:

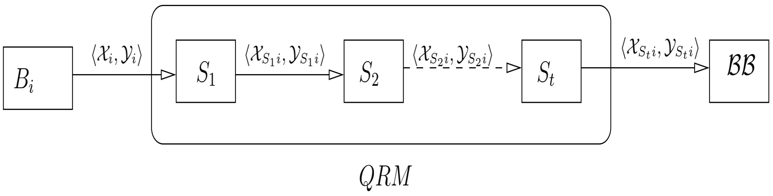
B_i
$\langle \mathcal{X}_i, \mathcal{Y}_i \rangle$
S_1
$\langle \mathcal{X}_{S_1 i}, \mathcal{Y}_{S_1 i} \rangle$
S_2
$\langle \mathcal{X}_{S_2 i}, \mathcal{Y}_{S_2 i} \rangle$
S_t
$\langle \mathcal{X}_{S_t i}, \mathcal{Y}_{S_t i} \rangle$
$\mathcal{BB}$
QRM

Fig. 2: Sequence of Sealing operation

$$
\begin{aligned}
X_{S_l i,(k,j)} &= g^{r_{S_l i,(k,j)}}.X_{S_{l-1} i,(k.j)} \\
&= g^{r_{S_l i,(k,j)}}.g^{(r_{i,(k,j)}+\sum_{t=1}^{l-1} r_{S_t i,(k,j)})} \\
&= g^{(r_{i,(k,j)}+\sum_{t=1}^{l} r_{S_t i,(k,j)})} \\
Y_{S_l i,(k,j)} &= \hat{r}_{S_l i,(k,j)}.h_A^{r_{S_l i,(k,j)}}.(h_{S/S_1,\ldots S_l})^{r_{S_l i,(k,j)}}.\left(X_{S_{l-1} i,(k,j)}\right)^{-x_{S_l}}.Y_{S_{l-1} i,(k,j)} \\
&= \hat{r}_{S_l i,(k,j)}.h_A^{r_{S_l i,(k,j)}}.(h_{S/S_1,\ldots S_l})^{r_{S_l i,(k,j)}}. \\
&\quad \prod_{t=1}^{l-1} \hat{r}_{S_t i,(k,j)}.h_A^{(r_{i,(k,j)}+\sum_{t=1}^{l-1} r_{S_t i,(k,j)})}.(h_{S/S_1\ldots S_l})^{(r_{i,(k,j)}+\sum_{t=1}^{l-1} r_{S_t i,(k,j)})}.G_\square \\
&= \prod_{t=1}^{l} \hat{r}_{S_t i,(k,j)}.h_A^{(r_{i,(k,j)}+\sum_{t=1}^{l} r_{S_t i,(k,j)})}.(h_{S/S_1\ldots S_l})^{(r_{i,(k,j)}+\sum_{t=1}^{l} r_{S_t i,(k,j)})}.G_\square
\end{aligned}
$$

After t sealing operation the bid-vector is reduced to

$$
\begin{aligned}
X_{S_t i,(k,j)} &= g^{(r_{i,(k,j)}+\sum_{l=1}^{t} r_{S_l i,(k,j)})} \\
Y_{S_t i,(k,j)} &= \prod_{l=1}^{t} \hat{r}_{S_l i,(k,j)}.h_A^{(r_{i,(k,j)}+\sum_{l=1}^{t} r_{S_l i,(k,j)})}.(h_{S/S_1\ldots S_t})^{(r_{i,(k,j)}+\sum_{l=1}^{t} r_{S_l i,(k,j)})}.G_\square \\
&= \prod_{l=1}^{t} \hat{r}_{S_l i,(k,j)}.h_A^{(r_{i,(k,j)}+\sum_{l=1}^{t} r_{S_l i,(k,j)})}.G_\square
\end{aligned}
$$

Algorithm 6: ZK1$(B_i, G_i, g_y, h_{B_i}, g)$

```
begin
    Bidder $B_i$ selects $a, b \in_R \mathbb{Z}_p$ and computes $\alpha = g^a$, $\beta = g_y^b$. Bidder $B_i$ sends $\alpha$ and $\beta$ to the auctioneer
    Auctioneer selects $c \in_R \mathbb{Z}_p$ and sends to $B_i$
    Bidder $B_i$ computes $r = a + cx_{B_i}$ and sends to the auctioneer
    Auctioneer verifies
        $g^r \overset{?}{=} \alpha.h_{B_i}^c$    (3)
        $g_y^r \overset{?}{=} \beta.G_i^c$    (4)
    if (relation 3 & 4 are TRUE) then
        Returns TRUE
end
```

ZK protocol

Zero-Knowledge (ZK) protocol [7] is a tool by which the prover can prove to another party (the verifier) that a function has been correctly computed, without revealing the secret parameters of the computation. The auction mechanism uses

Algorithm 7: ZK2$(B_i, QUAL, w)$

```
begin
    Bidder $B_i$ compute $\hat{R}_i = \prod_{k=0}^{d-1} \hat{r}_{i,(k,w_k)}$ and sends to the auctioneer
    All sealer $S_l \in QRM$ computes $\hat{R}_{S_l} = \prod_{k=0}^{d-1} \hat{r}_{S_l i,(k,w_k)}$ and $R_{S_l} = \sum_{k}^{d-1} r_{S_l i,(k,w_k)}$ and sends to the auctioneer
    Auctioneer computes
        $$\mathrm{X}_i = \prod_{k=0}^{d-1} X_{S_t i,(k,w_k)} \cdot \left(g^{\sum_{l=1}^{t} R_{S_l}}\right)^{-1} = g^{\sum_{k=0}^{d-1} r_{i,(k,w_k)}}$$
        $$\mathbb{G} = \prod_{k=0}^{d-1} G_{i,(k,w_k)} \cdot \left(\hat{R}_i\right)^{-1} = G_i^{\sum_{k=0}^{d-1} r_{i,(k,w_k)}}$$
    Bidder $B_i$ and auctioneer execute $ZK1(B_i, \mathbb{G}, G_i, \mathrm{X}_i, g)$
end
```

the ZK protocol to determine the winning bidder. Let $w = w_{d-1} \dots w_0$ be the winning price and B_i responds as the winner. The bidder B_i have to prove the following:

- B_i publishes $G_i = g_y^{x_{B_i}}$ and proves that G_i and h_{B_i} having common exponent (x_{B_i}) over g_y and g respectively, without disclosing the secret x_{B_i}. Algorithm 6 describes the proof.
- For $k = 0, 1, \dots d-1$, B_i publishes the product of all $\hat{r}_{i,(k,w_k)}$ and proves that he knows the common exponents over $X_{i,(k,w_k)}$s and $G_{i,(k,w_k)}$s. The proof would not be carried on individual items but exercised on the product of all $X_{i,(k,w_k)}$ (for $k = 0, 1, \dots d-1$). The Algorithm 7 describes the proof.

Does **ProcSwap()** vulnerable

The subprocess $EBY()$ is a recursive process that partitions the list L into two halves and invokes the **ProcSwap()**. The Fig 3 shows the process of partitioning ans swapping operation. $EBY()$ divides the list into some stacks of sealed bids. Every stack contains only two sealed bids where at least one of them must contains the *Yes Mark* on the P_{k,w_k} index. However, **ProcSwap()** procedure takes a stack

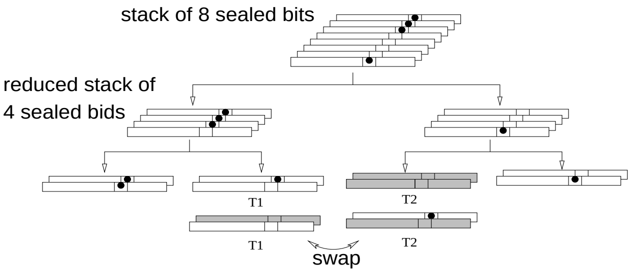

Fig. 3: Process of $EBY()$

(size 2) and demands additional information to determine the bid containing the *Yes Mark*. We claim that the additional information that is published in order to execute **ProcSwap()** does not compromise the receipt-freeness property.

Lemma 1. *Let a, b, c & $d \in \mathbb{Z}_p$ such that;*

$$\begin{aligned} a.b &= k_1 \qquad & c.d &= k_2 \\ a.c &= k_3 \qquad & b.d &= k_4 \end{aligned}$$

Though the values of k_1, k_2, k_3 & k_4 are known, it is computationally infeasible to find the unique solution of a, b, c & d.

Proof. In the above set of equation, any one of the equation is derivable from the other three equations. Let $a.b = k_1$, $c.d = k_2$ and $a.c = k_3$ are given, the fourth equation can be derivable from the given three equations, that is, $b.d = (a.b).(c.d).(a.c)^{-1} = k_1.k_2.k_3^{-1}$. Therefore the above system is effectively consists of three equations with four unknown variables. Henceforth infeasible to determine the unique solution of the a, b, c & b. If p is sufficiently big any random search is inefficient to get the solution of $a, b, c,$ & d □.

Let $T1$ be a stack containing two bids B_1 and B_2. Also let $T2$ be another stack containing two void bids V_1 and V_2. Therefore the $\mathcal{BB}$ already contains the values

$$k_1 = \hat{r}_{S_1B_1,(k,j)}.\hat{r}_{S_1B_2,(k,j)}$$

$$k_2 = \hat{r}_{S_1V_1,(k,j)}.\hat{r}_{S_1V_2,(k,j)}$$

(The procedure **ProcS**$(T1, S_1, P_{k,j})$ and **ProcS**$(T2, S_1, P_{k,j})$ publish the values) The call to the procedure **ProcSwap(T1,T2)** demands

$$k_3 = \hat{r}_{S_1B_1,(k,j)}.\hat{r}_{S_1V_2,(k,j)}$$

$$k_4 = \hat{r}_{S_2V_1,(k,j)}.\hat{r}_{S_1V_2,(k,j)}$$

Knowing the values k_1, k_2, k_3 & k_4 adversary would not able to resolve the secrets $\hat{r}_{S_1B_1,(k,j)}$ and $\hat{r}_{S_1B_2,(k,j)}$ without better than any random guess.