Application of Distributed Key Generation in Secured Sealed-Bid Auction Mechanism

Sanjit Kumar Roy
Roll No. 11/IT/415
Under the guidance of
Jaydeep Howlader
Assistant Professor

Department of Information Technology

National Institute of Technology
Durgapur, India

A Thesis submitted in partial fulfillment for the Degree of

Masters of Technology in
Information Technology

May 2013

Certificate Of Approval

The forgoing thesis is hereby approved as a creditable study of Technological subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite with degree for which it has been submitted. It is to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for the purpose for which it has been submitted.

Board of Thesis Examineer

-
-
-
-

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degrees or diplomas of the university or other institutes of higher learning, except which due acknowledgment has been maid in this text.

Signature

Certificate

This is certified that the work contained in this thesis entitled, **Application of Distributed Key Generation in Secured Sealed-Bid Auction Mechanism**, *by* **SANJIT KUMAR ROY**, *Roll No. 11/IT/415, has been carried out under the supervision of the undersign and this work has not been submitted elsewhere for any other degree.*

(Signature of Project Guide)
Jaydeep Howlader
Information Technology

(Signature of HoD)
Dr. Debashis Nandi
Information Technology

Acknowledgements

I hereby wish to express my sincere gratitude and respect to Assistant Proof. Jaydeep Howlader, Department of Information Technology, National Institute of Technology, Durgapur, under whom I had proud privilege to work. His valuable guidance and encouragement have really led me to the path of completion of this project. Any amount of thanks would not be enough for the valuable guidance of my supervisor. It was thus that working under him expertise and receiving a part of his tremendous knowledge became rewarding experience.

I would also like to thank all the faculty member of Information Technology deptartment for their devoted help and such a friedly environment in the department.

Finally, I would like to pen down my gratitude towards my parent and one of my friend Projit Sanyal for their continuous support and encouragement. It would have not been possible to complete my work without their support.

(Signature of Student)

Abstract

Sealed-Bid auctions are subject to bid-rigging attack by the coercer. Receipt-free sealed-bid auction mechanisms are developed to prevent bid-rigging. The prior receipt-free schemes assume the availability of untappable channel between the bidders and the auction authorities. However, it is often difficult and impractical to deploy untappable channel in real senario. Moreover they also assume the authorities not to be colluded, therefore no partial information of any bidder's secret is revealed. In this work we present a receipt-free sealed-bid auction scheme where neither the untappable channel is used nor we assume all authorities to be honest (there must be some honest authorities). We design secure multi-party computation to provide receipt-freeness, whereas deniable encryption followed by anonymous communication is used to relax the requirement of untappable channel. *Distributed Key Generation* is used for secure multiparty computation. We assume a set of player executes the protocol to output a common public key, where the secret key is shared among the players.

Keywords: Bid-rigging, Receipt-freeness, Incoercibility

Contents

List of Figures

Chapter 1

Introduction

Auction is an efficient and convincing method to establish the price of goods and trading the goods in the open market. In sealed-bid auction the bidders submit their bids securely in closed envelop. The bids are remain secure until they are not opened at the time of *opening*. During the *opening* the sealed envelopes are broken and the bids are disclosed. The winning price and winner is determined accordingly. Implementing the sealed-bid auction in the electronic domain, there are many threats to the security of the system [6, 3] like: correctness, fairness, privacy, nonrepudiation etc. Any adversarial activity that either corrupts the bids or manipulates the output of the auction is mostly due to the naive implementation of the system. The adversary may be the *insider* or *outsider*. The misbehavior of the *insider* (e.g. colluded auctioneers who involved in the conspiracy with coercer) yields an improper auction where the coercing bidder takes the advantage over others. Whereas the *outsider* attempts to corrupt the bids (sealed bids) which yield to the failure of the system. The system is also in trouble when the winner repudiates. There have been a number of schemes proposed for secure electronic sealed-bid auction [26, 22, 24, 11, 25, 2]. Those scheme do not address the problem of bid-rigging. Bid-rigging is the problem where the powerful entity called coercer orders the other to bid at low price so that the coercer could win the auction by quoting unreasonably low price (little higher that the other bidders). When bid-rigging happens the auction fails to meet the true valuation of the goods. The first proposal to prevent rigging was [18] where they introduced the receipt-free mechanism in an *Electronic Voting Protocol*. Receipt-freeness is the inability of any entity to prove his secret value in a voting or auctioning scenario. Most of the electronic sealed-bid auction, the bidder either publishes or carries a commitment of his secret bid. The commitment plays the role of receipt and is exploited by the coercer to determine the secret, even if the bidder is not willing to disclose.

1.1 Survey

The first receipt-free sealed-bid auction was proposed by *Abe, Suzuki* in [1]. Their scheme was base on secret-sharing over untappable channel. In their scheme the bidder constructs n shares of his bid and securely communicates the shares to the auctioneers over untappable channel. However the scheme fails to provide receipt-freeness in presence of dishonest (colluded) auctioneers. *Huang et al.* [17] proposed some improvement of *Abe, Suzuki's* [1] scheme, but could not overcome the problem of dishonest auctioneer. *Chen, Lee* [5] proposed another receipt-free scheme based on homomorphic encryption. In their proposed scheme, bidder along with seller construct the receipt-free bid over the untappable channel. Bidder and seller individually prove the validity of their operation. *Chen et al.* argued that the seller would not be colluded due to benefit collision. However the assumption is partially correct, as the seller may also be colluded when the item to be sold is not the property of the seller (e.g. government of a country wants to auction the mine sector). More over the auctioneer may open the bids and determines the highest bid price before the scheduled opening period. Whereas *Her et al.*[12] proposed another scheme of receipt-freeness, where the bidder has to do registration prior to bidding. Nevertheless, if the Registering authority is dishonest, the scheme fails to provide receipt-freeness. However *Gao et al.* [8] in their proposed scheme used undeniable signature for providing incoercibility in Electronic-Auction. The scheme demand all the bidders to be present during opening. Later on *Howlader, Ghosh, Pal* [14] proposed a receipt-free scheme for sealed bid auction using multiple sealer and single auctioneer. In their scheme, the bidder's secret bid is sealed by multiple sealers to form the receipt-free bid.

However the scheme fails to provide receipt-freeness as the verification carries the receipt of the initial secret bid.

In *Howlader et al.* [16, 15] deniable encryption [4, 13] is used as a tool for replacing the untappable channel, which was an essentially required in most of the prior receipt-free mechanism. However the technique fails to prevent receipt-freeness in the presence of dishonest authorities [15]. Latter on a "coercing resistant MIX" was proposed for anonymous bidding. In this case, the authorities receive anonymous bids from the bidder, hence could not corresponds the bidder "who-bid-what". On the other hand, the bidder transmits deniable ciphers, where he can plausibly deny his true value.

Chapter 2

Mathematical Concepts

In this chapter, we introduce the basic mathematical concepts which are used in our work.

2.1 Number Theory

2.1.1 Set

In this section, we describe the fundamental discrete structure on which all other discrete structures are built, namely, the set. Sets are used to group objects together. Often, the objects in a set have similar properties. An *axiomatic definition* of set is given below:
Definition: A set is an unordered collection of objects.
Definition: The objects in a set are called the *elements*, or *members*, of the set. A set is said to *contain* its elements. Set can be represented in various way.
Example:

1. The set of *natural numbers* $\mathbf{N} = \{\ 0, 1, 2, 3, \ldots\}$

2. The set of *integers* $\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

3. The set of *positive integers* $\mathbf{Z}^+ = \{1, 2, 3, \ldots\}$

4. The set of *rational numbers* $\mathbf{Q} = \{p/q \mid p \in \mathbf{Z}, q \in \mathbf{Z} \text{ and } q \neq 0\}$

5. The set of *vowel* $\mathbf{V} = \{a, e, i, o, u\}$

2.1.2 Group

Definition: A *group* $(G, \oplus)$, is a set of elements with a binary operation denoted by $\oplus$ that associates to each ordered pair (a, b) of elements in G an element $(a \oplus b)$ in G, such that the following axioms are obeyed:

1. **Closure:** If a and $b \in G$, then $a \oplus b \in G$.
2. **Associative:** $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ for all $a, b, c \in G$.
3. **Identity element:** There exists a (unique) elemnet $e \in G$ such that $e \oplus a = a \oplus e = a$ for all $a \in G$. The element e is called the *identity* of G.
4. **Inverse element:** For each $a \in G$, there exists a (unique) element $b \in G$ such that $a \oplus b = b \oplus a = e$. The element b is called the *inverse* of a.

Order of a group: The *order of a group* G is the cardinality or number of element in that group G.
Order of an element: The *order of an element* a in group G is the least positive integer k such that a^k is the identity.

2.1.3 Abelian Group

A group $(G, \oplus)$ is called *abelian* or *commutative* group if $a \oplus b = b \oplus a$ for all $a, b \in G$.

2.2 Cyclic Group

A group is **Cyclic** if every element of G is a power g^k (k is an integer) of a fixed element $g \in G$.

2.2.1 Generator:

Let $g \in \mathbb{Z}_n^*$, the order of g is the least positive integer i such that $g^i \equiv 1 \bmod n$. g will be called a generator of $\mathbb{Z}_n^*$, if i, the order of g equals to $\phi(n)$, where $\phi(n)$ is the cardinality of the set $\mathbb{Z}_n^*$.

2.3 Ring

A *ring* R, sometimes denoted by $\{R, +, \times\}$, is a set of elements with two binary operations, called *addition* and *multiplication*, such that for all $a, b, c \in R$ the following axioms are obeyed.

1. R is an abelian group with respect to addition.
2. **Closure under multiplication:** If a and b belong to R, then ab is also in R.
3. **Associativity of multiplication:** $a(bc) = (ab)c$ for all a, b, c in R.
4. **Distributive laws:** $a(b + c) = ab + ac$ for all a, b, c in R.
   $(a + b)c = ac + bc$ for all a, b, c in R.

2.4 Field

A *field* F, sometimes denoted by $\{F, +, \times\}$, is a set of elements with two binary operations, called *addition* and *multiplication*, such that for all $a, b, c \in F$ the following axioms are obeyed.

1. F is a ring.
2. **Commutativity of multiplication:** $ab = ba$ for all $a, b \in R$.
3. **Multiplicative identity:** There is an element 1 in R such that $a1 = 1a = a$ for all a in R.
4. **No zero divisors:** If a, b in R and $ab = 0$, then either $a = 0$ or $b = 0$.
5. **Multiplicative inverse:** For each a in F, except 0, there is an element a^{-1} in F such that $aa^{-1} = (a^{-1})\,a = 1$.

2.5 Intractable Mathematical Problems

2.5.1 Discrete Log Problem (DLP)

Let G, be a finite cyclic (multiplicative) group with cardinality n and a generator g. Given an element $a \in G$, find an integer x (or *the* integer x with $0 \leq x \leq n-1$) such that $a = g^x$ in G. Finding the unique integer x is hard and x is the discrete logarithm $\log_g a$.

Three different types of groups are commonly used for cryptographic applications: the multiplicative group of a finite field, the group of rational points on an elliptic curve over a finite field and the jacobian of a hyperelliptic curve over a finite field. Here we used DLP over finite fields.

2.5.2 Integer Factorization Problem (IFP)

Given the product n as a product of two distinct prime integers, it is computationally hard to determine the prime factors of n.

2.6 Cryptographic Algorithm (Asymmetric Cryptography)

A form of cryptosystem in which encryption and decryption are performed using two different keys, one of which is referred to as the public key and another one is referred to as the private key.
Public Key: The public key is made public by the entity and used in conjunction with a corresponding private key.
Private Key: The private key is the secret key and only known to the entity

Following are two asymmetric encryption decryption algorithms:

2.6.1 The RSA Public-key Encryption Algorithm

The *Ron Rivest, Adi Shamir and Leonard Adleman* (*RSA*) algorithm [21] is based on *integer factorization problem*. *RSA* algorithm are of three parts,

- Key Generation (Algorithm 2.1).
- Encryption (Algorithm 2.2).
- Decryption (Algorithm 2.3).

Algorithm 2.1: RSA Key Generation

Input: A bit length l.

Output: A random RSA key pair

Steps:

1 Generate two different random primes p and q each of bit length l.

2 $n := pq$.

3 Choose an integer e coprime to $\phi(n) = (p-1)(q-1)$.

4 $d := e^{-1}(mod\phi(n))$.

5 Return the pair (n, e) as the public key and the pair (n, d) as the private key.

Algorithm 2.2: RSA Encryption

Input: The RSA public key (n, e) of the recipient and the plaintext message $m \in \mathbb{Z}_n$.

Output: The ciphertext message $c \in \mathbb{Z}_n$.

Steps:

1 $c := m^e (\text{mod } n)$.

Algorithm 2.3: RSA Decryption

Input: The RSA private key (n, d) of the recipient and the ciphertext message $c \in \mathbb{Z}_n$.

Output: The recovered plaintext message $m \in \mathbb{Z}_n$.

Steps:

1 $m := c^d (\text{mod } n)$.

2.6.2 The ElGamal Public-key Encryption Algorithm

The *ElGamal* public key encryption algorithm [7] is based on *discrete log problem* (DLP) 2.5.1. The algorithm has three parts,

- Key Generation (Algorithm 2.4).
- Encryption (Algorithm 2.5).
- Decryption (Algorithm 2.6).

Let p is a prime number and G is the cyclic group of order $p-1$. Let $g \in G$ is a generator of the group.

Algorithm 2.4: ElGamal Key Generation

Input: G, g

Output: A random ElGamal key pair.

Steps:

1 Generate a random integer $d, 2 \leq d \leq k-1$.

2 Return $h = g^d$ as the public key and d as the private key.

Algorithm 2.5: ElGamal Encryption

Input: (G, g) and the ElGamal public key h, the plaintext message $m \in G$.

Output: The ciphertext message $(r, s) \in H \times G$ (where $H = \langle g \rangle$).

Steps:

1 Generate a (random) session key $d', 2 \leq d' \leq k-1$.

2 $r := g^{d'}$.

3 $s := mh^{d'}$.

Unlike *RSA*, *ElGamal* is a random encryption, that is if the message m is encrypted with the same public key for multiple times, ElGamal cipher differs every time as the element d' is selected randomly.

Algorithm 2.6: ElGamal Decryption

Input: (G, g) and the ElGamal private key d, the ciphertext message $(r, s) \in H \times G$.

Output: The recovered plaintext message $m \in G$.

Steps:

$m := sr^{-d}$.

Chapter 3

Secret Sharing

3.1 How To Share a Secret

Shamir's secret sharing is the threshold scheme, which is used to share a secret within more than one party. A secret s is shared within n party such that k party can recompute the secret s again, where $k \leq n$.

3.1.1 Methodology

The scheme is based on *polynomial interpolation:* given n points in the 2-dimentional plane $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ with distinct x_i's, there is one and only one polynomial $q(x)$ of degree $k-1$ such that $q(x_i) = y_i$ for all i.
Example: 2 points are sufficient to define a straight line, 3 points are sufficient to define a parabola, 4 points to define a cubic curve and so on.

3.1.2 Problems

In Shamir's secret sharing protocol [23] a trusted party (dealer) shares a secret $s \in \mathbb{Z}_p$ among the parties $P_1, \ldots, P_n$ in the following way. The dealer chooses at random a polynomial $f(x)$ over $\mathbb{Z}_p$ of degree $k-1$, such that $f(0) = s$. He then secretly transmits to each party P_i a share $s_i = f(i) \bmod p$. It is clear that $k-1$ or less parties have no information about the secret while k can easily reconstruct it by polynomial interpolation.

3.1.3 Note to Distributed Secret Sharing

In the presence of malicious adversary, shamir's secret sharing protocol could fail. Indeed, it is possible for a dealer to share values which do not lie on a polynomial of degree $k-1$. Also dishonest parties may contribute incorrect shares at reconstruction time. *Verifiable secret sharing (VSS)* protocols [19] are intended to prevent this possibility.
However this scheme approach us to *distributed secret sharing*, where every party act as a dealer and distribute shares to others of it's secret.

Chapter 4

Distributed Key Generation

4.1 Introduction

Distributed key generation is a main component of threshold cryptosystems. It allows a set of n servers to generate jointly a pair of public and private keys according to the distribution defined by the underlying cryptosystem without ever having to compute, reconstruct, or store the secret key in any single location and without assuming any trusted party (dealer). While the public key is output in the clear, the private key is maintained as a (virtual) secret shared via a threshold scheme. For discret-log-based (dlog-based) schemes, distributed key generation amounts to generating a secret sharing of a random, uniformly distributed value x and making public the value $y = g^x$.

4.2 Pedersen's Verifiable Secret Sharing

The scheme uses the parameters p and q, are the two large primes such that q divides $p-1$, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q, and g is a generator of G_q. Another additional element $h \in G_q$. It is assumed that the adversary cannot compute $\log_g h$. Let dealer will distribute it's secret $s \in \mathbb{Z}_q$. The scheme as follows:

1. Dealer publishes a commitment to $s : E_0 = E(s, t)$ for a randomly chosen

$t \in \mathbb{Z}_q$, where

$$E(s,t) = g^s h^t$$

2. Dealer chooses $F \in \mathbb{Z}_q[x]$ of degree at most $k-1$ satisfying $F(0) = s$, and computes $s_i = F(i)$ for $i = 1, \ldots, n$.
   Let $F(x) = s + F_1 x + \cdots + F_{k-1} x^{k-1}$. Dealer chooses $G_1, \ldots, G_{k-1} \in \mathbb{Z}_q$ at random and uses G_i when committing to F_i for $i = 1, \ldots, k-1$.

3. Let $G(x) = t + G_1 x + \cdots + G_{k-1} x^{k-1}$ and let $t_i = G(i)$ for $i = 1, \ldots, n$. Then dealer sends (s_i, t_i) secretly to P_i for $i = 1, 2, \ldots, n$.

4. Dealer compute,

$$\begin{aligned} E_j &= E(F_j, G_j) \\ &= g^{F_j} h^{G_j} \end{aligned}$$

   and broadcast E_j, for $j = 1, \ldots, k-1$.

When P_i has received his share (s_i, t_i) he verifies that

$$E(s_i, t_i) \stackrel{?}{=} \prod_{j=0}^{k-1} E_j^{i^j} \bmod p \tag{4.1}$$

If a party P_i holds a share that does not satisfy equation 4.1 then he *complains* against the dealer. The dealer reveals the share (s_i, t_i) matching equation 4.1 for each complaining party P_i. If any of the revealed shares fails this equation, the dealer is disqualified.

4.3 Pedersen Threshold Cryptosystem

Pedersen shown how the private key x is distributed such that any k or more members can find it. The scheme uses the parameters p and q, are the two large primes such that q divides $p-1$, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q, and g is a generator of G_q. The dealer P_i distributes x_i as follows:

1. P_i chooses at random a polynomial $f_i(z) \in \mathbb{Z}_q(z)$ of degree at most $k-1$ such that $f_i(0) = x_i$. Let
$$f_i(z) = f_{i0} + f_{i1}z + \cdots + f_{i,k-1}z^{k-1}$$
where $f_{i0} = x_i$.

2. P_i computes $F_{ij} = g^{f_{ij}}$ for $j = 0, \ldots, k-1$ and *broadcasts* $(F_{ij})_{j=1,\ldots,k-1}$ ($F_{i0} = h_i$ is known beforehand).

3. When everybody have sent these $k-1$ values, P_i sends $s_{ij} = f_i(j)$ *secretly* and a signature on s_{ij} to P_j for $j = 1, \ldots, n$ (in particular P_i keeps s_{ii}).

4. P_i verifies that the share received from, $P_j(s_{ji})$ is consistent with the previously published values by verifying that
$$g^{s_{ji}} \stackrel{?}{=} \prod_{l=0}^{k-1} F_{jl}^{il}$$
If this fails, P_i broadcasts that an error has been found, publishes s_{ij} and the signature and then stops.

5. P_i computes his share of x as the sum of all shares received in step 3 $s_i = \sum_{j=1}^{n} s_{ji}$.

4.4 Distributed Key Generation (DKG)

Distributed Key Generation (DKG) protocol allows a set of Players to generate a pair of public, private key. The public key is output in clear and the corresponding private key is shared among the players with (t-n) secret sharing [23]. Unlike Shamir's secret sharing [23], DKG does not assume any trusted party (Dealer) for compute and reconstruct the shares and secret respectively. DKG protocol successfully outputs the desire keys even there is an adversary who can corrupt at most $t-1$ Players to executes the protocol as instructed by the adversary. The secret sharing is verifiable if the protocol meets the following two requirements:

1. After receiving a share s_{ij} from Player P_i, the receiver P_j must be able to verify whether or not the share is a valid piece of the secret of P_i.

2. There is no perceivable advantage of determining the secret by randomly selecting any less than t number of shares.

Torben Pryds Pedersen [20] first proposed the verifiable DKG for discrete log base cryptosystem. The basic idea of the protocol is as follows:

- Players $\{P_1, P_2, \ldots, P_n\}$ with their randomly selected secret $\{s_1, s_2, \ldots, s_n\}$ respectively, initiate the protocol.
- Every Player P_j receives the share of the secret of P_i as s_{ij} and computes its secret as $x_j = \sum_{i=1}^{n} s_{ij}$.
- The public key is output as, $h = g^{\sum_{i=1}^{n} s_i}$.
- Decryption would be done by a quorum of at least $t > n/2$ Players.

4.4.1 Pedersen's Distributed Key Generation Protocol

Pedersen's protocol is verifiable where every Player checks whether or not the received shares are the correct pieces of the sender's secret. If the verification determines a Player as *cheater*, the protocol *disqualifies* the Player and the public key is output without the component of the *cheating* Player. The protocol is as follows:

1. Each Player P_i takes a random $t-1$ degree polynomial $f_i(x) = \sum_{k=0}^{t-1} a_{ik} x^k$ over $\mathbb{Z}_p^*$ where $a_{i0} = s_i$.
2. Player P_i computes $E_{ik} = g^{a_{ik}}$, for $k = 0, 1, \ldots, t$ and broadcasts E_{ik}.
3. After all players has broadcast their E_{ij}, every Player P_i computes the share of his secret for player P_j as $s_{ij} = f_i(j)$ and secretly transmits the share to Player P_j.
4. Player P_j verifies the share received from P_i is consistent by checking

$$g^{s_{ij}} = \prod_{k=0}^{t-1} (E_{ik})^{j^k} \mod p$$

If Player P_j computes a failure of the P_i's share, he complains against P_i. To Player P_i reveals his shares s_{ij} against every complains to satisfy the verification otherwise disqualified.

5. After having a defined set of qualifying Players ($QUAL$), the public key is determined as $h_s = \prod_{i \in QUAL} E_{i0}$

6. Player P_i sets his secret share as $x_i = \sum_{j \in QUAL} s_{ji}$

Pedersen's DKG protocol allows the adversary to bias the output key (public & private) to a non-uniform distribution. As the value of the private secret depends on the definition of $QUAL$ and the adversary can see all the broadcast information beforehand, it is shown in [10] that the adversary who controls some Players to react after seeing all the public information such that the Player may either *qualifies* or *disqualifies*. In fact Pedersen's DKG determines the $QUAL$ after outputting the keys. Later on [9, 10] proposed a new DKG based on the Pedersen's scheme.

4.4.2 New Distributed Key Generation

In the new DKG, the $QUAL$ is determines first then the private and public values are generated. The new DKG is as follows:

1. **Determining $QUAL$ and private secret:**

   (a) Each Player P_i randomly chooses two polynomials of degree $t-1$ as $f_i(x) = \sum_{j=0}^{t-1} a_{ij} x^j$ and $\acute{f}_i(x) = \sum_{j=0}^{t-1} \acute{a}_{ij} x^j$. Let $a_{i0} = s_i$. Player P_i broadcasts $E_{ik} = g^{a_{ik}} h^{\acute{a}_{ik}} \mod p$, for $k = 0, 1, \ldots, t-1$.

   (b) Player P_i computes the share $(s_{ij} = f_i(j), \acute{s}_{ij} = \acute{f}_i(j))$ and secretly sends to Player P_j.

   (c) Player P_j verifies the consistency of the P_i's share as

   $$g^{s_{ij}} h^{\acute{s}_{ij}} \stackrel{?}{=} \prod_{k=0}^{t-1} (E_{ik})^{j^k} \mod p$$

   Player P_j complains against P_i if the check fails.

(d) Player P_i broadcasts $(s_{ij}, \acute{s}_{ij})$ if there is a complain launched by P_j. The Player P_i disqualifies if either (1) there is more than $t-1$ complains against P_i or (2) $(s_{ij}, \acute{s}_{ij})$ reviled by P_i is falsify.

(e) The $QUAL$ is determined with the qualifying set of Players.

(f) Each Player P_i computes his secret share $x_i = \sum_{j \in QUAL} s_{ji}$

2. **Determining the public key using Pedersen's DKG**

(a) Each Player $P_{i \in QUAL}$, broadcast $C_{ik} = g^{a_{ik}} \mod p$, for $k = 0, 1, \ldots, t-1$.

(b) Each Player $P_{j \in QUAL}$, verifies the values broadcast by other Players $P_{i \in QUAL}$ as:

$$g^{s_{ij}} \stackrel{?}{=} \prod_{k=0}^{t-1} (C_{ik})^{j^k}$$

P_j complains against Player P_i if the check fails for Player P_i and publicly announces the $s_{ij}, \acute{s}_{ij}$.

(c) If the complain against P_k is **valid**, the $QUAL$ is reconfigured as $QUAL = QUAL - P_k$.

(d) The Players $P_i \in QUAL$ recompute their private secret $x_i = \sum_{j \in QUAL} s_{ji}$. The public key is output as $h_s = \prod_{i \in QUAL} C_{i0}$ for $i \in QUAL$

Chapter 5

Application of Disributed Key Generation (DKG) in Sealed-Bid Auction

5.1 Introduction

Sealed-Bid auctions are subject to bid-rigging attack by the powerful entity called coercer. Bid-rigging is the problem where coercer orders the other to bid at low price so that the coercer could win the auction by quoting unreasonably low price (little higher that the other bidders). When bid-rigging happens the auction fails to meet the true valuation of the goods. Receipt-free sealed-bid auction mechanisms [14] are developed to prevent bid-rigging. Receipt-freeness is the inability of any entity to prove his secret bid. The [14] shceme is based on multi-party computation and the entities are *bidder, sealer* and *auctioneer*. The mechanism is with a group of n sealers. At the time of sealing every sealer is needed to seal. The mechanism fails if any sealer is unavailable. We use a threashold cryptosystem $(k-n)$ on sealers. The protocol will be successfull in the preasence of any k sealers where $k \leq n$.

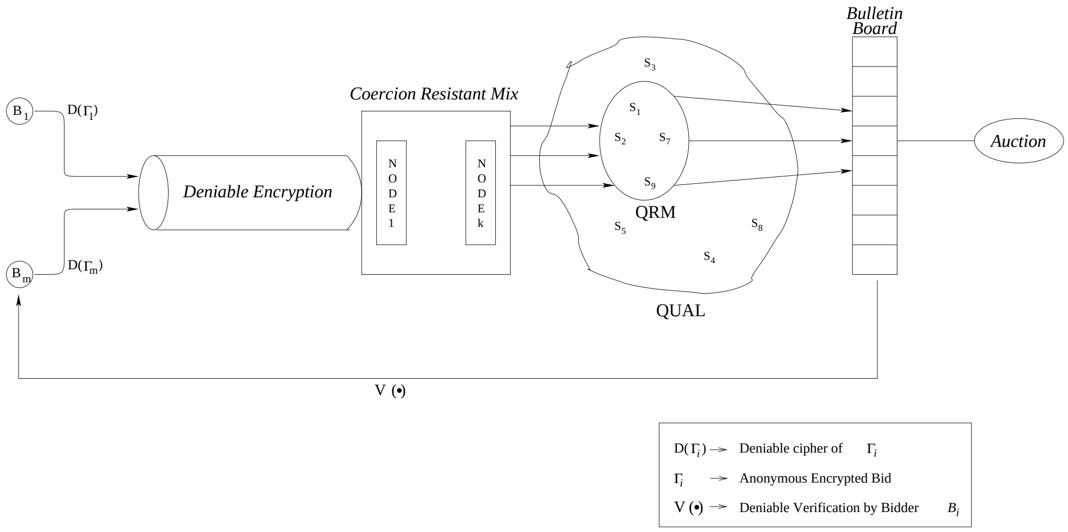

Figure 5.1: Setting of Sealed-Bid Auction

5.2 Problem and Parameters

The proposed receipt-free mechanism is based on multi-party computation where the entities are as follows:

Bidder who bids. We assume there are m number of bidders and represent the set of bidders as $B = B_1, B_2, \ldots, B_m$. We also assume that the bidders bid honestly.

Sealer acts as an authority who performs the sealing operation to form receipt-free bid. There finite number of sealers and we assume that as n. The sealer set is represented as $S = S_1, S_2, \ldots, S_n$ There may some colluded sealers.

Auctioneer computes and determines the winning price, during opening. The winner proves his winning price to the auctioneer. If the winning bidder not responds, the auctioneer along with the sealers determines the repudiating bidder.

Coercer is the powerful entity who indulge bid-rigging. We assume the coerce to be powerful enough to demand all the keys and randomness of the bidders after the bidding phase. Coercers can intercept the communication at any point of time. We also assume the existence of some colluded authorities who may leak some information that yield coercing. However, we assume that coercer does not able to control the bidders at the time of their casting.

We consider the setting as shown in the figure 5.1. The bidder B_i computes his secret bid vector as $\mathit{\Gamma}_i$ and reencrypts with *Deniable Encryption* to form a deniable cipher. The deniable cipher are communicated to the sealer *(QRM) Coercion Resistant Mix* [15]. The deniable encryption followed by anonymity prevent the coercers to coerce the bidder.

If the coercer eavesdrop the channel to capture the secret bid, *Deniable Encryption* allows the bidders to produce a fake bid and get ried of coercing. On the other hand the colluded authorities could convey a secret bid to the coercer, but could not corresponds the bidders with their bids as the *Mix* produces anonymous output.

5.3 Protocol

5.3.1 System Setting

Let p and q denote large primes such that q divides $p-1$, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q, and $g \in \mathbb{Z}_p^*$ is an element of order q. Following are the description of the key pairs of different entities.

- Bidder B_i's private key is x_{B_i} and public key is $h_{B_i} = g^{x_{B_i}}$.
- Auctioneer A's private key is x_A and public key $h_A = g^{x_A}$.
- k Sealers execute the Distributed Key Generation (DKG) 4.4 protocol and output a set of qualifying sealers $QUAL$ 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 $QRM \subseteq QUAL$ is able to seal the bidder's 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 $x_{S_i} = \lambda_{ij} x_i$ where λ_{ij} [1] is the Lagrange interpolation coefficient.
- Sealer $S_i \in QRM$ publishes his public key for sealing as $h_{S_i} = g^{x_{S_i}}$.
- We denote $h_{S/S_1,S_2,...,S_r} = h_S(h_{S_1} h_{S_2} \ldots h_{S_r})^{-1}$.

[1]Lagrange interpolation coefficient for the $i^{\,th}$ point is $\lambda_{ij} = \prod\limits_{i \neq j} \frac{x - x_j}{x_i - x_j}$

- $g_y \in \mathbb{Z}_p^*$ is a generator indicates '*YES Mark*'.
- A price list $P = \{p_1, p_2, \ldots, p_n\}$ in ascending order is published by the auctioneer.

5.4 Receipt-free sealed-bid auction mechanism

The proposed receipt-free mechanism consists of three phase: *Bidding*,*Opening* and *Trading*.

5.4.1 Bidding Phase:

Bidding operation is performed in three steps:

- Constructing the encrypted bid vector,
- Sealing the bid-vector,
- Bid verification.

5.4.1.1 Constructing the encrypted bid vector

Each bidder B_i decides his bidding price $p_j \in P$ and computes the encrypted bid vector as,

$$\begin{aligned} {}_i\Gamma_j \quad &=({}_iX_j, {}_iY_j) \\ &= \begin{cases} g^{{}_ir_j}, h_A^{{}_ir_j}.h_S^{{}_ir_j}.({}_iG_y)^{{}_ir_j} & \text{if } p_j = j^{th} \text{ price} \in P \\ g^{{}_ir_j}, h_A^{{}_ir_j}.h_S^{{}_ir_j} & \text{otherwise} \end{cases} \end{aligned}$$

for every price $1 \leq j \leq n$, B_i randomly select ${}_ir_j \in_R \mathbb{Z}_p^*$ and computes the bid vector. Bidder B_i puts his '*Signed YES Mark*' at the j^{th} price as $({}_iG_y)^{{}_ir_j}$, where ${}_iG_y = g_y^{x_{B_i}}$. The operator '.' is multiplication in $\mathbb{Z}_p^*$. B_i sends the encrypted bid vector to a predefined $QUAL$ over a "Coercing Resistant Mix".

5.4.1.2 Sealing the bid-vector

We assumed a $QUAL$ is predefined. Sealing would be performed by any $k/2 < t \leq k$ number of sealers from the $QUAL$. The set of such t sealers is called quorum QRM. Without losing any generality, consider that QRM consists of t sealers $\{S_1, S_2, ..., S_t\}$. It is not mandatory to execute the sealing operation sequentially, but for the simplification we assume that sealing is done sequentially (e.g. S_1 followed by S_2 followed by S_3 etc.)

1. After receiving encrypted bid-vector $\langle {}_i\Gamma_j \rangle$, sealer S_1 partially seal the bid-vector as $\langle {}_i\Gamma_{j,S_1} \rangle$, where ${}_i\Gamma_{j,S_1} = \{{}_iX_{j,S_1}, {}_iY_{j,S_1}\}$ and

$$\begin{aligned} {}_iX_{j,S_1} &= g^{{}_ir_{j,S_1}} . {}_iX_j \\ &= g^{{}_ir_j + {}_ir_{j,S_1}} \\ {}_iY_{j,S_1} &= {}_ir_{j,S_1} \; . \; g^{{}_ir_{j,S_1}} . h_A^{{}_ir_{j,S_1}} . h_{S/S_1}^{{}_ir_{j,S_1}} . \left({}_iX_j\right)^{-x_{S_1}} . {}_iY_j \\ &= {}_ir_{j,S_1} . g^{{}_ir_{j,S_1}} . h_A^{{}_ir_{j,S_1}} . h_{S/S_1}^{{}_ir_{j,S_1}} . h_{S_1}^{-{}_ir_j} . h_S^{{}_ir_j} . h_A^{{}_ir_j} . {}_iG_j \\ &= {}_ir_{j,S_1} . g^{{}_ir_{j,S_1}} . h_A^{{}_ir_j + {}_ir_{j,S_1}} . h_{S/S_1}^{{}_ir_{j,S_1}} . h_{S/S_1}^{{}_ir_j} . {}_iG_j \\ &= {}_ir_{j,S_1} . g^{{}_ir_{j,S_1}} . h_A^{{}_ir_j + {}_ir_{j,S_1}} . h_{S/S_1}^{{}_ir_j + {}_ir_{j,S_1}} . {}_iG_j \end{aligned}$$

   for every price $1 \leq j \leq n$, sealer S_1 randomly selects ${}_ir_{j,S_1} \in_R \mathbb{Z}_p^*$. We denote ${}_iG_j = \{1, ({}_iG_y)^{{}_ir_j}\}$. Sealer S_1 forwards the partially sealed bid to the next sealer S_2.

2. After receiving the partial sealed bid-vector $\langle {}_i\Gamma_{j,S_1} \rangle$, sealer S_2 further sealed the bid-vector as $\langle {}_i\Gamma_{j,S_2} \rangle$, where ${}_i\Gamma_{j,S_2} = \{{}_iX_{j,S_2}, {}_iY_{j,S_2}\}$ and

$$\begin{aligned} {}_iX_{j,S_2} &= g^{{}_ir_{j,S_2}} . {}_iX_{j,S_1} \\ &= g^{{}_ir_j + \sum_{l=1}^{2} {}_ir_{j,S_l}} \\ {}_iY_{j,S_2} &= {}_ir_{j,S_2} \; . \; g^{{}_ir_{j,S_2}} . h_A^{{}_ir_{j,S_2}} . h_{S/S_1,S_2}^{{}_ir_{j,S_2}} . \left({}_iX_{j,S_1}\right)^{-x_{S_2}} . {}_iY_{j,S_1} \\ &= {}_ir_{j,S_2} . g^{{}_ir_{j,S_2}} . h_A^{{}_ir_{j,S_2}} . h_{S/S_1,S_2}^{{}_ir_{j,S_2}} . h_{S_2}^{-\left({}_ir_j + {}_ir_{j,S_1}\right)} . {}_ir_{j,S_1} . g^{{}_ir_{j,S_1}} \\ &\quad . h_A^{{}_ir_j + {}_ir_{j,S_1}} . h_{S/S_1}^{{}_ir_j + {}_ir_{j,S_1}} . {}_iG_j \\ &= \left(\prod_{l=1}^{2} {}_ir_{j,S_l}\right) . g^{\sum_{l=1}^{2} {}_ir_{j,S_l}} . h_A^{{}_ir_j + \sum_{l=1}^{2} {}_ir_{j,S_l}} . h_{S/S_1,S_2}^{{}_ir_j + \sum_{l=1}^{2} {}_ir_{j,S_l}} . {}_iG_j \end{aligned}$$

for every price $1 \leq j \leq n$, sealer S_2 randomly selects ${}_ir_{j,S_2} \in_R \mathbb{Z}_p^*$. Sealer S_2 forwards the partially sealed bid to S_3 and so on. In this way the last sealer S_t receives the partial sealed bid-vector $\langle {}_i\Gamma_{j,S_{t-1}} \rangle$.

3. After receiving partial sealed bid-vector $\langle {}_i\Gamma_{j,S_{t-1}} \rangle$, sealer S_t finally sealed the bid-vector as $\langle {}_i\Gamma_{j,S_t} \rangle$, where ${}_i\Gamma_{j,S_t} = \{{}_iX_{j,S_t}, {}_iY_{j,S_t}\}$ and

$$\begin{aligned}
{}_iX_{j,S_t} &= g^{{}_ir_{j,S_t}}.{}_iX_{j,S_{t-1}} \\
&= g^{{}_ir_j + \sum_{l=1}^{t} {}_ir_{j,S_l}} \\
{}_iY_{j,S_t} &= {}_ir_{j,S_t} \ . \ g^{{}_ir_{j,S_t}}.h_A^{{}_ir_{j,S_t}}.h_{S/S_1,S_2,\ldots,S_t}^{{}_ir_{j,S_t}}.\left({}_iX_{j,S_{t-1}}\right)^{-x_{S_t}}.{}_iY_{j,S_{t-1}} \\
&= \Big(\prod_{l=1}^{t} {}_ir_{j,S_l}\Big).g^{\sum_{l=1}^{t} {}_ir_{j,S_l}}.h_A^{{}_ir_j + \sum_{l=1}^{t} {}_ir_{j,S_l}}.h_{S/S_1,S_2,\ldots,S_t}^{{}_ir_j + \sum_{l=1}^{t} {}_ir_{j,S_l}}.{}_iG_j \\
&= \Big(\prod_{l=1}^{t} {}_ir_{j,S_l}\Big).g^{\sum_{l=1}^{t} {}_ir_{j,S_l}}.h_A^{{}_ir_j + \sum_{l=1}^{t} {}_ir_{j,S_l}}.{}_iG_j
\end{aligned}$$

for every price $1 \leq j \leq n$, sealer S_t randomly selects ${}_ir_{j,S_t} \in_R \mathbb{Z}_p^*$.

At least $k/2+1$ sealer have to perform the sealing to form the receipt-free sealed-bid.

5.4.2 Bid Verification

After performed the sealing by QRM, the sealed bid is published in the public board. Receipt-freeness does not allow the bidders to prove bidding values, but allows the bidders to verify whether their bids have been correctly sealed or not. The verification of bid is done with the response attached by the sealers with the bid vector.

1. The first sealer S_1 computes the response of the bidder B_i's sealing as ${}_iR_{S_1}$,

where

$$
\begin{aligned}
{}_iR &= {}_iX_1 . {}_iX_2 \ldots {}_iX_n \\
&= g^{\sum_{j=1}^{n} {}_ir_j} \\
{}_iR_{S_1} &= \Big(\prod_{j=1}^{n} {}_ir_{j,S_1}\Big) . g^{\sum_{j=1}^{n} {}_ir_{j,S_1}} . {}_iR \\
&= \Big(\prod_{j=1}^{n} {}_ir_{j,S_1}\Big) . g^{\sum_{j=1}^{n} {}_ir_{j,S_1}} . g^{\sum_{j=1}^{n} {}_ir_j} \\
&= \Big(\prod_{j=1}^{n} {}_ir_{j,S_1}\Big) . g^{\sum_{j=1}^{n} \left({}_ir_j + {}_ir_{j,S_1}\right)}
\end{aligned}
$$

and forwards to the next sealer S_2.

2. Sealer S_2 computes his response as,

$$
\begin{aligned}
{}_iR_{S_2} &= \Big(\prod_{j=1}^{n} {}_ir_{j,S_2}\Big) . g^{\sum_{j=1}^{n} {}_ir_{j,S_2}} . {}_iR_{j,S_1} \\
&= \Big(\prod_{j=1}^{n} \prod_{l=1}^{2} {}_ir_{j,S_l}\Big) . g^{\sum_{j=1}^{n} \left({}_ir_j + \sum_{l=1}^{2} {}_ir_{j,S_l}\right)}
\end{aligned}
$$

and forwards to S_3.

3. Thus after t sealing the response is computed as,

$$
\begin{aligned}
{}_iR_{S_t} &= \Big(\prod_{j=1}^{n} {}_ir_{j,S_t}\Big) . g^{\sum_{j=1}^{n} {}_ir_{j,S_t}} . {}_iR_{S_{t-1}} \\
&= \Big(\prod_{j=1}^{n} \prod_{l=1}^{t} {}_ir_{j,S_l}\Big) . g^{\sum_{j=1}^{n} \left({}_ir_j + \sum_{l=1}^{t} {}_ir_{j,S_l}\right)}
\end{aligned}
$$

and published on the public board.

For every bidder B_i, Auctioneer A computes ${}_i\mathbb{X}$ for $i = 1, 2, ..., m$ where,

$$
\begin{aligned}
{}_i\mathbb{X} &= \left(\prod_{j=1}^{n} {}_iX_{j,S_t}\right)^{x_A} \\
&= h_A^{\sum_{j=1}^{n} \left({}_ir_j + \sum_{l=1}^{t} {}_ir_{j,S_l}\right)}
\end{aligned}
$$

and writes on public board.

Bidder B_i checks, validity of his bid-vector as follows:

1. Bidder B_i computes ${}_iC$ where,

$$\begin{aligned} {}_iC &= \prod_{j=1}^{n} {}_iY_{j,S_t} \cdot {}_i\mathbb{X}^{-1} \\ &= \left(\prod_{j=1}^{n} \prod_{l=1}^{t} {}_ir_{j,S_l} \right) \cdot g^{\sum_{j=1}^{n} \sum_{l=1}^{t} {}_ir_{j,S_l}} \cdot {}_iG_j \end{aligned}$$

2. Bidder B_i verifies

$$ {}_iC \cdot {}_iR \stackrel{?}{=} {}_iR_{S_t} \cdot {}_iG_j $$

If bidder B_i fails to verify his sealed bid, it raises a complain.

5.4.3 Opening Phase

After successfully executing the bidding phase, the bids are opened as per the scheduled time. The bids are opened in descending order (for highest price Auction). That is auctioneer first open the n^{th} price, if no one bids, then open the $(n-1)^{th}$ price and so on until there is '*YES Mark*' for the j^{th} price. Opening of j^{th} price is as follows:

1. All sealers $S_l \in QUAL$ compute V_{j,S_l}, where,

$$ V_{j,S_l} = \prod_{i=1}^{m} \left({}_ir_{j,S_l} \cdot g^{{}_ir_{j,S_l}} \right) $$

and send to auctioneer A until the winning price is determined.

2. (a) Auctioneer A computes $\mathbb{V}_j$ and $\mathbb{Y}_j$ where,

$$\begin{aligned}
\mathbb{V}_j &= \prod_{l=1}^{t} V_{j,S_l} \\
&= \prod_{l=1}^{t}\prod_{i=1}^{m} \left({}_i r_{j,S_l}.g^{{}_i r_{j,S_l}}\right) \\
&= \left(\prod_{i=1}^{m}\prod_{l=1}^{t} {}_i r_{j,S_l}\right).g^{\sum_{i=1}^{m}\sum_{l=1}^{t} {}_i r_{j,S_l}} \\
\mathbb{Y}_j &= \prod_{i=1}^{m}\left\{{}_i Y_{j,S_t}.({}_i X_{j,S_t})^{-x_A}\right\} \\
&= \left(\prod_{i=1}^{m}\prod_{l=1}^{t} {}_i r_{j,S_t}\right).g^{\sum_{i=1}^{m}\sum_{l=1}^{t} {}_i r_{j,S_l}}.\prod_{i=1}^{m} {}_i G_j
\end{aligned}$$

(b) Auctioneer A compute $\mathbb{G}_j$ where,

$$\begin{aligned}
\mathbb{G}_j &= \mathbb{Y}_j.(\mathbb{V}_j)^{-1} \\
&= \prod_{i=1}^{m} {}_i G_j
\end{aligned}$$

(c) Auctioneer A declear j^{th} price as the winning price if,

$$\mathbb{G}_j \neq 1$$

5.4.4 Trading:

After declaration of winning price, bidder B_i claims his victory with his '*YES Mark*' (${}_iG_y$) and '*Signed YES Mark*' (${}_iG_w$). Let w is the winning price.

1. In case of single winner auctioneer A verifies the winner,

$$\mathbb{G}_j \stackrel{?}{=} {}_iG_w$$

   If the verification succeed, auctioneer declares B_i as winner.

2. However for multiple winner, auctioneer A and bidders $B_i \in ClaimedWinner$ individually execute two *interactive zero-knowledge proof*s to satisfy,

(a) ${}_iG_y$ and h_{B_i} has common exponent,

(b) ${}_iX_w$ and ${}_iG_w$ has common exponent.

3. After successful executing the *zero-knowledge proof* by multiple bidders, auctioneer A verifies the following:

$$\prod_{\substack{i=claimed\\winners}} {}_iG_w \stackrel{?}{=} \mathbb{G}_j$$

If the verification succeed, auctioneer declares the claimed bidders as winners.

Chapter 6

Analysis of Receipt-Free Sealed-Bid Auction

6.1 Receipt-Freeness

The quorum (QRM) receives the encrypted bids anonymously. Therefore, even the recipient of any encrypted bids is colluded (i.e. intend to convey the encrypted bids ${}_i\Gamma_j$ to the coercer), then also the coercing is insignificant as the colluded entity could not resolve "who-encrypts-what". On the other hand, the sealed-bids are receipt-free as the sealing operation engraves randomness of the sealers $S_l \in QRM$. The scheme guarantees receipt-freeness if at least one of the sealer in QRM is honest.

Deniable Verifiability: Bidder has to compute ${}_iC$ during verification of his sealed bid. In that case bidder need not to disclose the position of the '*YES Mark*'. For example, let bidder B_i has '*YES Mark*' on j^{th} price but during the verification he can plausibly deny and show that '*YES Mark*' is in some $i \neq j^{th}$ position.

Bidder can prove to *coerer*, that the '*signed YES Mark*' is at i^{th} position, as the marks are in product form.

6.2 Non-Repudiation

The auction scheme determine the winning price, not the winner. So winning-bidder may keep silent (repudiate). The auction scheme is able to identify the winning-bidder in case of repudiation. The procedure as follows:

Let w^{th} price is the winning price. Auctioneer asks all sealers (member of QRM) to write the initial encrypted bid-vector for w^{th} price, that is $\langle {}_i\Gamma_w \rangle$ on the public board. Every sealer $S_l \in QRM$, computes $h^{{}_ir_w}_{S_{l \in QRM}} = {}_iX_w^{x_{S_l}}$ (where x_{S_l} is the sealing key of sealer S_l) and writes on the public board against $\langle {}_i\Gamma_w \rangle$. Auctioneer computes the 'Mark' of the bidder B_i as,

$$ {}_iG_w = {}_iY_w . \left\{ {}_iX_w^{x_A} \prod_{l \in QRM} h^{{}_ir_w}_{S_l} \right\}^{-1} $$

The mark would be either '*No Mark*' = 1 or '*Signed YES Mark*' = $({}_iG_y)^{{}_ir_w}$. The auctioneer asks all bidders to compute '*Signed YES Mark*' for w^{th} price, i.e, $({}_iG_y)^{{}_ir_w}$, and execute *interactive zero-knowledge proof*, to verify that $({}_iG_w)^{{}_ir_w}$ and ${}_iX_w$ has same exponent as ${}_ir_w$. If the marking of a bid $\langle {}_i\Gamma_w \rangle$ is $X \neq 1$ then there should be at least one bidder B_i for which the '*Signed YES Mark*' is also X. Thus the repudiating bidder could be identified.

6.3 Correctness

6.3.1 Publicly Verifiable Opening

Let auctioneer executes the opening and declare the j^{th} price as the winning price. Any one (insider/outsider) will be able to verify the winning price. Let an outsider P wants to verify the winning price. It will go through the following steps:

1. P will compute,

$$\begin{aligned}
\mathbb{PY}_j &= \prod_{i=1}^{m} {}_iY_{j,S_t} \\
&= \Big(\prod_{i=1}^{m}\prod_{l=1}^{t} {}_ir_{j,S_l}\Big).g^{\sum\limits_{i=1}^{m}\sum\limits_{l=1}^{t} {}_ir_{j,S_l}}.h_A^{\sum\limits_{i=1}^{m}\left({}_ir_j+\sum\limits_{l=1}^{t} {}_ir_{j,S_l}\right)}.\prod_{i=1}^{m} {}_iG_j \\
\mathbb{PX}_j &= \prod_{i=1}^{m} {}_iX_{j,S_t} \\
&= g^{\sum\limits_{i=1}^{m}\left({}_ir_j+\sum\limits_{l=1}^{t} {}_ir_{j,S_l}\right)}
\end{aligned}$$

from public board. P will ask to auctioneer to sign blindly on $\mathbb{PX}_j$ and let the value become,

$$_A\mathbb{PX}_j = h_A^{\sum\limits_{i=1}^{m}\left({}_ir_j+\sum\limits_{l=1}^{t} {}_ir_{j,S_l}\right)}$$

2. Now P gets V_{jS_l} is the composite randomness for all the j^{th} bids by sealer S_l, on the bulletin board and will compute,

$$\begin{aligned}
\mathbb{PV}_j &= \prod_{l=1}^{t} V_{j,S_l} \\
&= \prod_{l=1}^{t}\prod_{i=1}^{m} ({}_ir_{j,S_l}.g^{{}_ir_{j,S_l}}) \\
&= \left(\prod_{i=1}^{m}\prod_{l=1}^{t} {}_ir_{j,S_k}\right).g^{\sum\limits_{i=1}^{m}\sum\limits_{l=1}^{t} {}_ir_{j,S_k}}
\end{aligned}$$

3. Finally P will verify,

$$\mathbb{PY}_j.\,({}_A\mathbb{PX}_j)^{-1} \stackrel{?}{=} \mathbb{PV}_j$$

6.3.2 Adversary

Adversary may corrupt the bid vector. Here we assume that the bidders are honest during bidding. Moreover coercer would not gain any advantage by, corrupting bidder's bid vector due to the benefit collision. So only some insider (i.e,

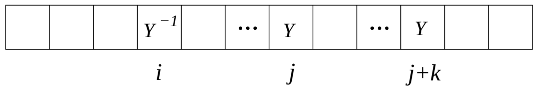

Figure 6.1: Bid Corruption

sealer) may corrupt bidder's bid. For example, let B_i's bid vector having the '*YES Mark*' at j^{th} position as shown in figure 6.1. An adversary may corrupt the bid by putting '*YES Mark*' at some $j + k^{th}$ position. At the same time adversary nullify his corruption by putting 'YES^{-1} *Mark*' as some other place.

At the time of verification bidder would not be able to verify that his bid vector has been corrupted. During opening, auctioneer will compute the $j + k^{th}$ price as the winning price. Auctioneer moves for *repudiation check*. As repudiation check does not able to identify any bidder, therefore *Auctioneer* moves for further opening. Therefore, any corruption only delays the opening, but can not victimize any bidder.

6.4 Efficiency

The efficiency of the receipt-free sealed-bid auction scheme depends on the bidding time and the sealing operation of the bidder and sealer respectively . The simulation result in Figure 6.2 shows the bidding time with verifying price range and key size 256, 512, 768 and 1024 bits. In each cases increasing the key size by 256 bits the bidding time increased by approximately three times in order.

The second simulation result Figure 6.3 shows the time required to perform a single sealing operation of a sealer with varying price range and varying key 256, 512, 768 and 1024. Like bidding time, the time for sealing operation is also increased by approximately three times in order.

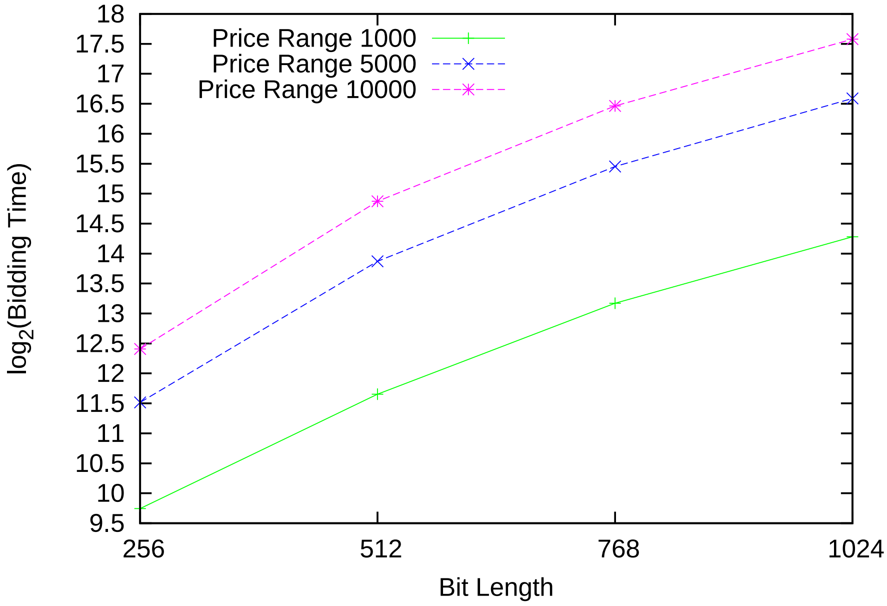

Figure 6.2: Bidding Time

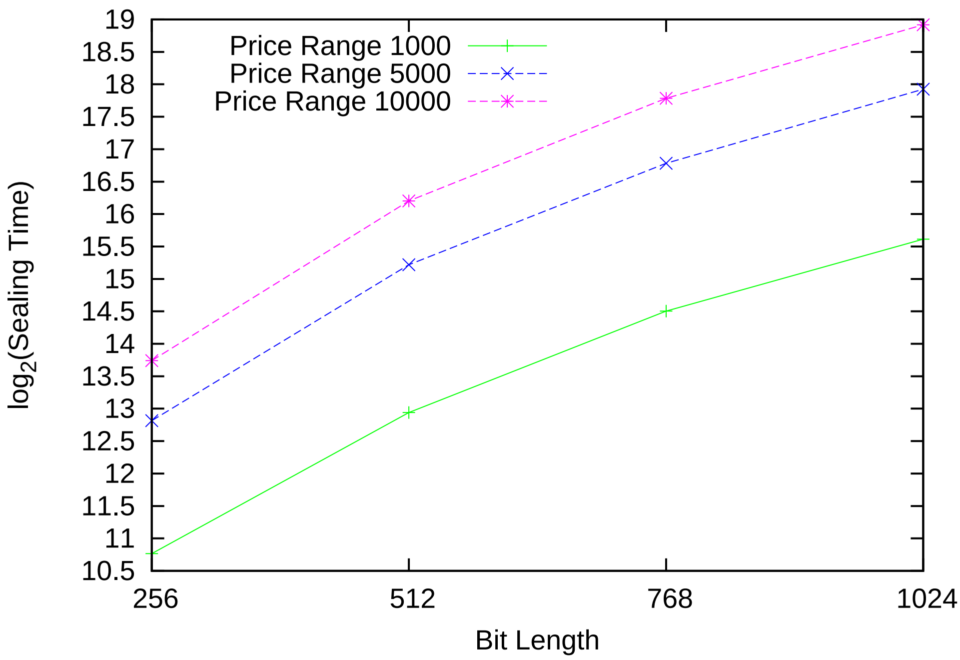

Figure 6.3: Sealing Time

6.5 Overhead of Message Passing for Misbehaviors in Distributed Key Generation

The DKG protocol executes in a broadcast manner. Therefore, there are many message exchange between the players. The number of messages increased as the number of player misbehaves. Here we analyze the overhead of message exchanging in context of misbehaving players. Let there are n players among them k players misbehaves. Now, k misbehaving players each may send false share to t other players where $t = 1, 2, \ldots, (n-1)$. When a player P_i communicates false share, then P_i broadcast $(n-1)$ complains against P_j. On the other hand P_j broadcast $(n-1)$ messages with his secrets share. So one misbehavior causes $2(n-1)$ message overhead. Therefore k misbehavior causes $2(n-1)k$ message overhead. In this sequel, if there are n players with k misbehaving players, where each k players misbehaves with t other then the total message overhead is $2(n-1)kt$.

Chapter 7

Conclusion

The proposed Receipt-free sealed bid auction scheme is based on multiple entity threshold trust model. The scheme involves multiple sealers and one auctioneer. An *qualifying* set of sealers ($QUAL$) is defined beforehand, by executing the *Distributed Key Generation* protocol within the sealers. A subset of the $QUAL$, called *Quorum* QRM, performs sealing of bids. The bidder uses the public key of the $QUAL$ to form his encrypted bid-vector, whereas the decryption key is shared among the members of QRM. Our proposed scheme guarantees the,

- Exemption of untappable channel.
- Receipt-freeness.
- Verifiability.
- Non-Repudiation.

7.1 Further Work

The proposed scheme defines a price list as a linear array of *Minimum* to *Maximum* price. The computation and space complexity is proportional to the length of the price list. For example if

- Let key size is 512 bit.

N	N	N	N	N	N	N	Y	N	N	10^3
N	N	N	Y	N	N	N	N	N	N	10^2
N	Y	N	N	N	N	N	N	N	N	10^1
N	N	N	N	N	N	N	N	N	Y	10^0

Figure 7.1: Decimal representation of bid vector.

- For price list 1000, bidding time is 3223 milisecond, and sealing time is 7856 milisecond.
- For price list 10000, bidding time is 29995 milisecond, sealing time is 75408 milisecond.

- Let key size is 512 bit.
  - For 512 bit key and 1000 price list, the bit vector would be 512*1000*2 bits.
  - For 10000 price list, the bid vector would be 512*10000*2 bits.

The scheme may be further improved if the price list is designed as *Multiple Decimal Price Vector (MDPV)*. In *MDPV* each vector represents one decimal position. For example the i^{th} vector represents 10^{i-1} decimal position. The example in 7.1 '*Yes Mark*' of 7319 is represented in *MDPV*. The top vector represents the 10^3 decimal position. That is the left most cell of the top vector represents 0×10^3, next cell represents 1×10^3 and so on. Therefore increase of the price list with a multiple of 10 results an increment of additional 10 entities in *MDPV*. The future work may comprise to modify the scheme to fit the idea in receipt-free sealed-bid auction.

References

[1] Masayuki Abe and Koutarou Suzuki. Receipt-free sealed-bid auction. In *ISC*, LNCS 2433, pages 191–199. Springer, 2002. 2

[2] 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. 1

[3] Colin Boyd and Wenbo Mao. *Security Issues for Electronic Auctions*. HP Laboratories technical report. Hewlett-Packard Laboratories, 2000. 1

[4] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *CRYPTO 97*, pages 90–104, 1997. 3

[5] Xiaofeng Chen, Byoungcheon Lee, and Kwangjo Kim. Receipt-free electronic auction schemes using homomorphic encryption. In *ICISC*, LNCS 2971, pages 259–273. Springer, 2003. 2

[6] Matthew K. Franklin and Michael K. Reiter. The design and implementation of a secure auction service. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 22(5):302–312, 1996. 1

[7] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985. 9

[8] Chongzhi Gao, Zheng an Yao, Dongqing Xie, and Baodian Wei. Electronic sealed-bid auctions with incoercibility. In *Electrical Power Systems and Computers*, LNEE 99, pages 47–54. Springer, 2011. 2

[9] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *EUROCRYPT*, LNCS 1592, pages 295–310. Springer, 1999. 17

[10] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems*. *Journal of Cryptology*, 20(1):51–83, 2007. 17

[11] Michael Harkavy, J. D. Tygar, and Hiroaki Kikuchi. Electronic auctions with private bids. In *Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce - Volume 3*, WOEC'98, pages 6–6. USENIX Association, 1998. 1

[12] Yong-Sork Her, Kenji Imamoto, and Kouichi Sakurai. Some remarks on security of receipt-free e-auction. In *ICITA (2)*, pages 560–563. IEEE Computer Society, July 2005. 2

[13] Jaydeep Howlader and Saikat Basu. Sender-side public key deniable encryption scheme. In *ARTCom*, pages 9–13. IEEE Computer Society, 2009. 3

[14] Jaydeep Howlader, Anushma Ghosh, and Tandra DebRoy Pal. Secure receipt-free sealed-bid electronic auction. In *IC3*, CCIS 40, pages 228–239. Springer, 2009. 2, 19

[15] Jaydeep Howlader, Jayanta Kar, and Ashis Kumar Mal. Coercion resistant mix for electronic auction. In *Information Systems Security*, LNCS 7671, pages 238–248. Springer, 2012. 3, 21

[16] Jaydeep Howlader, Vivek Nair, Saikat Basu, and A. K. Mal. Uncoercibility in e-voting and eauctioning mechanisms using deniable encryption. *International Journal of Network Security & Its Applications (IJNSA)*, 3(2):97–109, 2011. 3

[17] Zheng Huang, Weidong Qiu, Haibin Guan, and Kefei Chen. Efficient receipt-free electronic auction protocol. In *SITIS*, pages 1023–1028. IEEE Computer Society, 2007. 2

[18] Benaloh Josh and Tuinstra Dwight. Receipt-free secret-ballot elections (extended abstract). In *ACM symposium on Theory of computing*, STOC '94, pages 544–553, 1994. 1

[19] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, CRYPTO '91, pages 129–140. Springer-Verlag, 1991. 12

[20] Torben Pryds Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In *Advances in Cryptology-EUROCRYPT'91*, LNCS 547, pages 522–526. Springer-Verlag, 1991. 16

[21] Ron Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. 7

[22] Kouichi Sakurai and Shingo Miyazaki. A bulletin-board based digital auction scheme with bidding down strategy-towards anonymous electronic bidding without anonymous channels nor trusted centers. In *cryTEC1999*, pages 180–187, 1999. 1

[23] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. 11, 15

[24] Kapali Viswanathan, Colin Boyd, and Ed Dawson. A three phased schema for sealed bid auction system design. In *ACISP*, LNCS 1841, pages 412–426. Springer, 2000. 1

[25] Chia-Chi Wu, Chin-Chen Chang, and Iuon-Chang Lin. New sealed-bid electronic auction with fairness, security and efficiency. *J. Comput. Sci. Technol.*, 23(2):253–264, 2008. 1

[26] Hu Xiong, Zhiguang Qin, Fengli Zhang, Yong Yang, and Yang Zhao. A sealed-bid electronic auction protocol based on ring signature. In *ICCCAS*, pages 480–483. IEEE, 2007. 1