EFFICIENT IMPLEMENTATION AND COMPUTATIONAL ANALYSIS OF PRIVACY-PRESERVING PROTOCOLS FOR SECURING THE FINANCIAL MARKETS

by

Ramiro Alvarez

A Thesis Submitted to the Faculty of

The College of Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

Florida Atlantic University

Boca Raton, FL

August 2018

Copyright 2018 by Ramiro Alvarez

EFFICIENT IMPLEMENTATION AND COMPUTATIONAL ANALYSIS OF PRIVACY-PRESERVING PROTOCOLS FOR SECURING THE FINANCIAL MARKETS

by

Ramiro Alvarez

This thesis was prepared under the direction of the candidate's thesis advisor, Dr. Mehrdad Nojoumian, Department of Computer Science and Engineering, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:

Mehrdad Nojoumian, Ph.D.
Thesis Advisor

Taghi Khoshgoftaar, Ph.D.

Yufei Tang, Ph.D.

Erdoğ Nurgun, Ph.D.
Chair, Department of Computer Science and Engineering

Stella N. Batalama, Ph.D.
Dean, The College of Engineering and Computer Science

Khaled Sobhan, Ph.D.
Interim Dean, Graduate College

July 31, 2018
Date

ACKNOWLEDGEMENTS

To begin with, I would like to mention the immense gratitude I feel towards my thesis advisor Dr. Mehrdad Nojoumian. I am grateful for all the times Mehrdad provided insightful knowledge, and added constructive criticism. I am also grateful for his incredible patience, and support during difficult times. I am forever in your debt for opening the doors into the world of research.

Next, I would like to thank the committee members Dr. Taghi Khoshgoftaar and Dr. Yufi Tang who have accepted to review my thesis, and be present during my defense. In addition, I would like to thank all my mentors who have been a source of inspiration, and encouragement during this long track. I will always be grateful for all the wisdom, and knowledge that has been passed down.

My deep gratitude to the staff at Florida Atlantic University. I want the graduate advisor Jean Mangiaracina to know that I appreciate her guidance and willingness to always help eagerly. I would not have been at this step without the dedicated Florida Atlantic University staff.

Finally, I want to express my many thanks to my family, my wife, and friends. In particular, my wife deserves a special thanks. She has been my source of regeneration, and vitality during this difficult task. Thank you for motivating me to attend graduate school, and thank you for always believing in me.

ABSTRACT

Author:	Ramiro Alvarez
Title:	Efficient Implementation and Computational Analysis of Privacy-Preserving Protocols for Securing the Financial Markets
Institution:	Florida Atlantic University
Thesis Advisor:	Dr. Mehrdad Nojoumian
Degree:	Master of Science
Year:	2018

Auctions are a key economic mechanism for establishing the value of goods that have an uncertain price. In recent years, as a consequence of the ubiquitous emergence of technology, auctions can reach consumers, and as a result drive market prices, on a global scale. Collection of private information such as past trades exposes more information than desired by market participants. The leaked information can be statistically analyzed to provide auctioneers, or competitors, an advantage on future transactions. The need to preserve privacy has become a critical concern to reach an accepted level of fairness, and to provide market participants an environment in which they can bid a true valuation. This research is about possible mechanisms to carry out sealed-bid auctions in a distributed setting, and provides the reader with the challenges that currently persevere in the field. The first chapter offers an introduction to different kinds of auction, and to describe sealed-bid auction. The next chapter surveys the literature, and provides necessary theoretical background. Moving on to chapter 3, instead of solely focusing on theoretical aspects of sealed-bid auctions, this chapter dives into implementation details, and demonstrates through communication and computational analysis how different settings affect performance.

To my beloved wife Ile Marjan Taghioff, thank you for you love, and patience. You always push me to be a better version of myself

EFFICIENT IMPLEMENTATION AND COMPUTATIONAL ANALYSIS OF PRIVACY-PRESERVING PROTOCOLS FOR SECURING THE FINANCIAL MARKETS

List of Figures ix

1 Introduction 1

1.1 Introduction to Sealed-Bid Auctions 1

1.2 Desired qualities 3

1.3 Security Concerns 4

2 Survey on Privacy-Preserving Auctions 7

2.1 First Price Sealed-Bid 7

2.2 Vickrey Auctions 13

2.3 M+1 Auctions 18

2.4 Rule flexible Sealed-Bid Auctions 20

2.5 Combinatorial Auctions 22

3 Communication and Computation Analysis of Sealed-Bid Auctions 29

3.1 PROTOCOL DESCRIPTION 29

3.1.1 Hiroaki Kikuchi 30

3.1.2 Kazue Sako 31

3.1.3 Sakurai and Miyazaki 32

3.2 RESULTS AND DISCUSSION 33

3.2.1 Multiparty Computation 34

3.2.2 Public Key Encryption 36

3.2.3 Commitment Scheme 41
3.2.4 MPC vs Public Key Encryption vs Commitment Scheme . . . 43
3.3 Software Implementation Challenges 46
3.3.1 Big Integer Math and Prime Numbers 47
3.3.2 Concurrency in C++ 48

4 Concluding Remarks and Future Work 49

Appendices 52
A PUBLICATIONS: 53

Bibliography 54

LIST OF FIGURES

1.1 Auction Classification 2

1.2 Auction Properties 3

1.3 Auction Properties 4

2.1 Auction Model 8

3.1 Kazue Sako Auction Model 31

3.2 Sakurai and Miyazaki Auction Model 32

3.3 Initialization Time (Kikuchi MPC) 35

3.4 Verification Kikuchi MPC: 128 bits & 512 bits 36

3.5 Initialization (Kikuchi Veriable) 37

3.6 Verification time (Kikuchi Veriable) 38

3.7 Initialization (Sako ElGamal) 39

3.8 Initialization (Sako RSA) 39

3.9 Verifications Sako ElGamal 40

3.10 Verifications for Sako RSA 42

3.11 Initialization (Sakurai & Miyazaki) 43

3.12 Verifications Sakurai & Miyazaki 44

3.13 Initialization Times 45

3.14 Verifications Times 45

CHAPTER 1
INTRODUCTION

An auction is a mechanism useful for establishing the price of goods. In general, two groups seller(s) and bidders, trade commodities. Most auctions include an auctioneer(s), who is responsible for arranging the auction, accepting the bids, and declaring a winner on behalf of the seller.

In our present time, it is not unusual for financial transactions to take place without ever having to engage face to face with another person. Instead, we participate in economic transactions with nothing more than a computer and a credit card. As a result, electronic auctions are capable of registering many people, and are not limited by space or geolocation as traditional auctions. A challenge however, is in keeping auction fair even in the case bidders or auctioneers attempt to be dishonest.

Various cryptographic schemes are discuss in literature in order to securely implement auctions. The goal of these protocols is to prevent any of the participants from deviating from the protocol, and to provide fairness. The rest of this chapter will describe several auction types, security considerations, and possible forms of cheating.

1.1 INTRODUCTION TO SEALED-BID AUCTIONS

Auctions can take many forms depending on the methods employed for registering participants, bidding, and opening the bids. For instance, if an auction requires participants to bid in an increasing fashion, then it is said to fall under the category of English auctions. On the other hand, a Dutch auction, is an opposite approach, in which the standing price is iterative decrease until someone is willing to pay. Yet

another form, considered practical for electronic auctions, is a construct requiring the sealing of bids. The process is analogous to some degree to placing a letter inside an envelope to be open only by its recipient, which in the context of auctions involves participants sealing their bids, usually to be opened by an auctioneer.

In addition, we can consider the cases when the winner pays his own value or the case when he pays the price closest to his value, that is the bidder pays second highest value. In such cases, we label an auction as first price when the winner pays his own price, and state that the auction is second price when he must pay the second highest. It is worth knowing, that in literature, second-price sealed-bid auctions are commonly referred to as Vickrey auctions, named after Professor William Vickrey who is credited for formally describing the mechanism. As a visual aid, a tree structure is provided below.

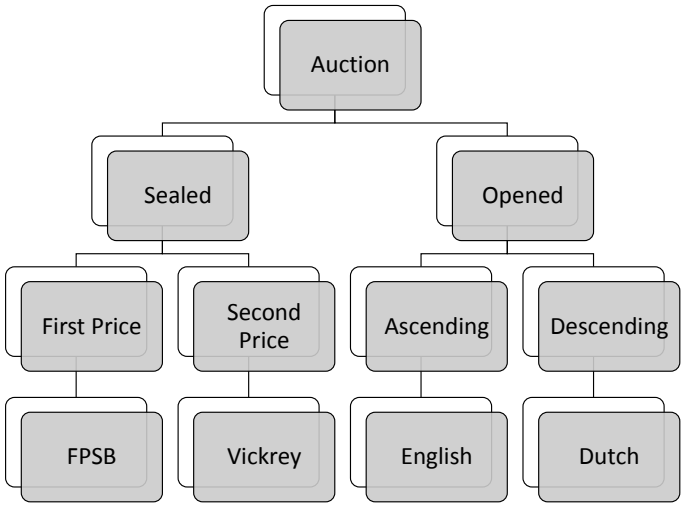


Figure 1.1: Auction Classification

1.2 DESIRED QUALITIES

When an auction takes place over the internet we face a number of challenges not present in auctions taking place physically. One of the advantages of an auction taking place in person is that we can associate a bid to a participant, which means no one can deny making a bid. However, there are disadvantages, such as number of people that can participate in this kind of auctions due to geolocation, and available space. With online auctions we can overcome the issue of location, and space, however we leave it up to the players to follow the auction protocols. Since the trust is placed in the hands of bidders, an auction method must ensure non-repudiation, robustness, and fairness, while concealing losing bids and providing secrecy and anonymity for bidders. At a minimum an auction should have the fundamental qualities described in [55]. Qualities of greater importance are expressed concisely in figure 2.

Secrecy	The identity of the bidder is never revealed.
Non-repudiation	A bidder should not be able to deny sending a bid which was truly submitted.
Verifiability	The winning bid should be recognized by all the members in the transaction as being the true winner by a process anyone can use to verify.
Anonymity	The protocol should not leave indications linking a bidder to a bid. In other words the bidder bid relation must be kept concealed.
Fairness	Under no circumstance should one of the players have an advantage over another player due.
Robustness	In the case of players willing to cheat, a counter strategy must exist such that it prevents those actions.
Losing Bid Privacy	Determination of the winning party of an auction should not arrive at the expense of opening or decrypting the losing bids.

Figure 1.2: Auction Properties

1.3 SECURITY CONCERNS

Unlike other systems, auctions consist of autonomous self-interested players that can form strategies in order to achieve personal goals. Therefore, design of a system with strategic players brings another layer of complexity to auction protocol. At a minimum, an auction mechanism must first arrive at the correct winner according to the auction rules, which typically are publicly known to all parties. Then, secondly, an exchange between the correct winner and the seller must take place according to the agreement established, for example, the winner pays the correct price to the seller, and the seller hands the item or service acquired by the bidder for his payment. However, providing a protocol which simply covers these two minimum requirements is not enough to provide the desired qualities previously mentioned in 1.2, and furthermore the system would be exposed to numerous abuses by players. Typical forms of cheating that can occur are described below.

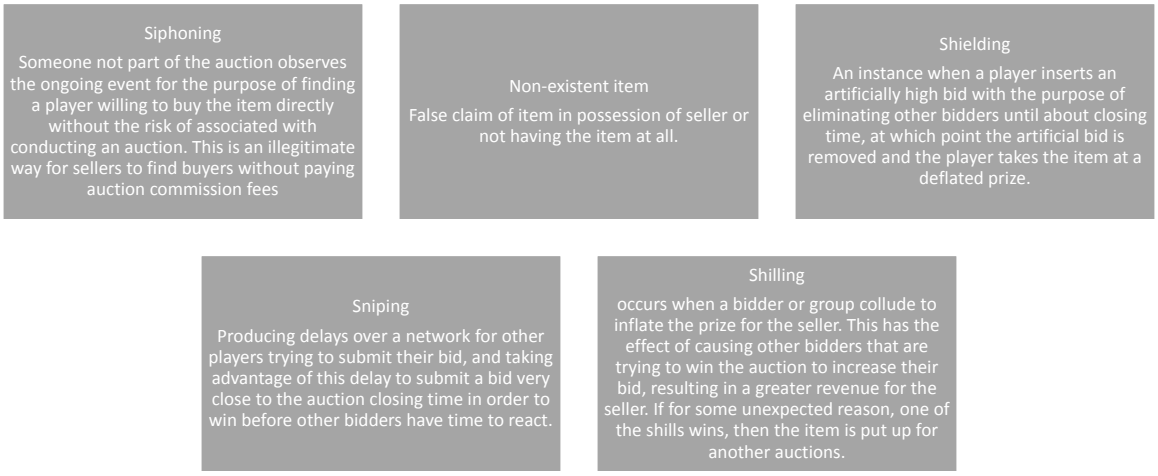


Figure 1.3: Auction Properties

In addition, to the type of misbehavior which can occur from sellers, and bidders, auctioneers can also be corrupted. One way an auctioneer can behave maliciously is by colluding with one of the players, and not accepting all submitted bids, but instead selectively picking out which bids should be considered so that a specific player takes the prize. Yet another unwanted outcome is that of an auctioneer colluding with a sellers to maximize profits. This can occurs in several ways, but a common method seen in second-price auctions is to submit an artificial bid as close as possible to the winning bid for the purpose of driving up the price paid. Overall, an auctioneer delineating from the rules is a major concern, which is central to some of the papers published in literature. These papers consider the ideas of third trusted parties (TTP), threshold trust, two-server trust, and distributed bidder trust. However, not all protocols are constructed with a corrupted auctioneer in mind due to computation complexity, security, and anonymity. Below is a summary of the possible trust models.

Auctioneer Trust: A naive approach for relying on the auctioneer to follow protocol as an honest agent of the process.

Distributed Bidder Trust: Bidder divide the trust among themselves, there is no auctioneer at all.

Trusted Third Parties: The auctioneers and bidders have a third party, with no personal gain from the auction which is trusted by both. This TTP can ensure that the important steps of the protocols are followed.

Threshold Trust: Consist of having more than one auctioneer. Auctioneers can only collude if a number of them are working together to harm the protocol. This number is the threshold, and as long as the number of corrupt auctioneers is less than this number the auction can remain fair.

Two-Server Trust: This method splits the trust among two entities. The auctioneers own one of the servers, and the bidders owned the other server. Protocol consistency is met as long as the two entities do not collide.

CHAPTER 2
SURVEY ON PRIVACY-PRESERVING AUCTIONS

The following section is an overview of important publications related to electronics auctions. For ease of reading, the section will be split into the following categories: FPSB, Vickrey, English, and Dutch auctions. For the most part we are interested in sealed-bid mechanisms.

2.1 FIRST PRICE SEALED-BID

The next section covers publications of auctions targeted at first price sealed-bid auctions. Similar to Vickrey, participants of this mechanism simultaneously send their bids in a sealed format. No bidder should know the content of another bid except his/her own. However, different from Vickrey, FPSB require that the winner pay his bid value. Some of the early works on the topic include the papers of [28], [12] [61], [60], and [68]. A brief description follows below.

Based on a foundation of secure function computation protocols, Kikuchi, Hakavy, and Tygar, create a first price sealed-bid auction. The auction protocol follows a model similar to Franklin and Reiter [19], in the sense that there exist one seller, multiple bidders, and multiple auctioneers (Figure 2 provides a visual aid for these type of common constructions for auctions). A set of prices k, are published during initialization. Bidders can have the option of assigning ID or zero to each price k depending on their valuation of the good. Once a bidder has prepare a sequence of bids for each k, each sequence becomes the input to a multiparty computation protocol (MPC). Addition provided with MPC determines the winner. For a price k, if only a

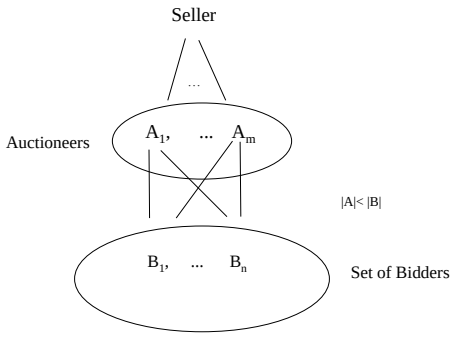


Figure 2.1: Auction Model

single winner exist, then the MPC reveals his ID, otherwise if multiple winners exist, the MPC reveals the sum of their ID. When no winner is found at a given price k, the result is zero. If a tie occurs which is likely for lower discrete k values, then subsequent rounds with different bid values are constructed with the winners from the previous round. The protocol can be improved by [29], providing an improvement on privacy at double the cost of computation, and it can also be improve as suggested by [54] to provide fairness, anonymity, and robustness. Finally, the authors of [13] created an alternative method for better efficiency, however, the authors construct a protocol using deniable signature. The protocol offers higher round complexity but overall improved efficiency, and better bandwidth. A feature of entertainment is added by [42].

Another general scheme applicable to first price auction rule is that presented by [12]. Bid privacy is achieved with homomorphic encryption, ϕhiding assumption, and multi-party computation. The protocol does not depend on circuit evaluation, however it requires the use of two servers, however, users communicate with only one of the servers, and the two servers communicate among themselves. Homomorphic public key encryption is used to seal the bids. Through a multiparty computation scheme

the semi-trusted trusted party compares the bids of two different bidders and determines the highest of the two values by comparing the encrypted bits starting from the most significant bit position. A result can be retrieved by the properties granted with the ϕ-hiding assumption. At the end of the protocol only the highest bidder is found, but it can be extended to find the second highest. One of the disadvantages of the protocol is that it con only function as intended with pairwise comparison at each step. For robust protocol each bidder must check the other bidder, that is A checks on B and B checks on A.

In [61] the proposed mechanism was using undeniable signature schemes. Bidders use undeniable signature to send their encrypted bid. For deciding a winner, the auctioneer begins at the highest price possible from the list, if a bidder meets the price from the list, then he must prove it with the undeniable signature, and the auctioneer opens the bid in order to publicly verify the result. If no one meets the price, the auctioneer goes on the next highest price, similar to a Dutch auction, until a winner is found.

By comparison, the author of [60] make use of public key cryptosystem. The construction can be build on public cryptosystems such as ElGamal and RSA. Essentially, a set of bid values exist which belong to the set V. During the bidding phase bidders encode a predetermine message M of their message and send it to an auctioneer. Auctioneers begin at highest bid value, and use the associated key to decrypt each received message. When a key opens a message, it follows the that message corresponds to the highest value. The main goal of the approach is to hide the bid value of losers with a protocol which guarantees that a bid cannot be successfully decrypted unless it is the highest bid, more specifically, a probabilistic encryption of a submitted bid that cannot be decrypted unless for sure it is the highest bid value. The only problem however, is that auctioneers hold the key and can decide to open every received message M. To circumvent this scenario the authors suggest sharing the key among

the auctioneers using secret sharing scheme.

Unlike the other protocols mentioned, the approach taken by [68] is the first approach to use only hash functions for a sealed-bid auction. More precisely the protocols utilizes an intractable hash function, such as SHA-1. The protocol is as follows. First, bidders compute a hash chain on their bid with the use of a randomly selected seed. The bid is sent along with a signature for non-repudiation purposes. During the opening phase, the auctioneer performs an equality test on a given price j. If no one satisfies the test of equality, then the auctioneer decreases the price by 1. The procedure is performed iterative until a price is found for which the test result is valid. As an alternative, the paper proposes that to avoid bidders communicating to the auctioneer when to open a specific bid, the protocol could be modified by sending the randomly chosen seed along with the message, and to avoid an auctioneer from learning the secret, the protocol should have multiple authorities which must come together to decrypt the message.

Another approach taken is the use of an off-line trusted third patry (TTP), with a dutch style approach, and key chaining are presented in a novel idea by . In this protocol bidders choose a random list of integers, and compute a list of α_{ij} where $\alpha = g^{s_{ij}} mod p$ and s_{ij} the list of random integers can only be decrypted by the TTP in case of verification. The list of integers is used as the input of the verifiable encryption VE(s_{ij}), and the output is send along the α value as tuple. The auctioneer receives the tuple in addition to a signature on that tuple. If the auctioneer can verify the received information, then a certificate is received by the particular bidder. In the final step of the registration the auctioneer publishes to a bulletin board the tuples as well as the ecryption key corresponding to the bidding prices. Entering the bidding phase, player send a concatenated result which includes information, such as the player choice on a specific price. Bidders have shares of the public key that decrypt the bid. The possible set of prices are opened from highest in a decreasing

order. Due to the property of universal verifiability, the price of the highest bidder is guaranteed to be the winner of the auction. Upon finding a winner, the auctioneer can open this bid, but cannot open the subsequent lower bids. In case of a dispute, or cheating bidder trying to disturb the protocol, the TTP can be brought to determine if a bidder is cheating or not, otherwise he remains offline reducing round complexity. The method of key chaining in this paper does not yields strong bid privacy for losing players relying on strong unrealistic trust that auctioneers and TTP will not collude to reveal the information, neither does it provide with unconditional privacy on the bids. With the modification presented in [53] both of these properties are achieved. Essentially the key chaining is modified such that finding the highest price completely breaks the key chain and bids can no longer be revealed.

On another approach approach [36], a new construction is achieved by the use of binding group signatures. For this protocol bidders must be awarded a certificate which permits them to participate in the auction. Then using the private membership, a particular bidder can place a bid and sign it with group signature. A group signature on a bid is distributed using group signature sharing scheme. During the opening phase the auctioneers retrieve the highest price using multi-party computation, and can find the identity associated with the bid by the revocation procedure of group signature scheme.

One more protocol [50] entails the usage of homomorphic secret sharing scheme. The new idea presented in this paper is to use a form of verifiable secret sharing (VSS), namely Pedersen's secret sharing scheme [47] to prevent sealed auction from attacks that typically arise in secret sharing schemes in the form of auctioneer-bidder-collusion(ABC), bidder-bidder-collusion (BBC), and a dispute attack. The constructions involves two rounds of communication. First, bidders commitment on a bid, and secondly bidders share opening information along with the auctioneers for determination of winner. In short, the protocol is as follows. A list of possible prices, the

whole set denoted by w already exist, or is predetermined. Each of the auctioneers establish corresponding Paillier's encryption function [44], and decryption key, while publishing onto a bulletin board the public encryption key, and encryption function. Bidders generate bidding vector by choosing an integer for each possible price. A non-zero random integer indicated participation by bidder, while a zero clearly indicates not wanting to buy at that certain price. Using system parameter established by the group of auctioneers, bidders commit the bid vectors. Binary search along with homomorphic secret recovery determine the winning bid.

An additional scheme proposes realizing an auction without auctioneers [74]. To remove auctioneers from the protocols, computations are handled by bidders. Each bidder generates a bid vector contain decision of bidding at a particular price. Each bidder slices and shares his vector with the other bidders keeping only a part secret. Each bidder add the received slice to his/her own vector. Assuming that no bidders share the same highest value, the seller can recover the winner by adding and subtracting bid vectors until he finds the index of the highest price. Finally, the seller can test the commitments to find a matching bid.

In the categories of auction utilizing homomorphic bid opening is [56]. The protocol is based on homorphic encryption. Each auctioneer chooses a private key and using threshold secret sharing shares the key among the other auctioneers. Bidders create a bid bit vector with "0" indicating opting out, and "1" indicating participating. The bid vector of each bidder is encrypted with ElGammal public key encryption, and for robustness a proof of logarithms is assessed. Similar to other homomorphic bid opening schemes, a binary search is computed until the winning price is found. Finally, all the bidders with a non-zero for the winning price in their bid vector are contestants for the prize.

The paper by [38, 37] describes a commitment scheme consist of a trusted initializer T and n number of bidders. During the initialization step, T selects n polynomials of

degree n-1 and sends g_i to P_i and also n-1 distinct points on each g_i to other players. Each player P_i computes $y_i = g_i(x_i)$ as a committed value and broadcasts y_i to other players, where x_i is the secret of P_i. Players maintain their bid in a binary a vector, and also a base 10 integer. In the reveal phase, the winner proves his claim by revealing commitments. Losers also prove that their bids have been less than the winning price. For the winner, players first investigate the validity of $y_i = g_i(x_i)$. They then check to see if all n-1 points are on $g_i(x)$.

Table 2.1: First Price Sealed-Bid Auctions Summary

Reference	**Cryptographic Method**	**Adversary Model**	**Security Model**
[28]	Relies on MPC, more specifically on addition property	Passive	Unconditional
[12]	Homomorphic Encryption and MPC	Active	Computational
[61]	A bid is an Undeniable Signature	Passive	Computational
[60]	ElGammal or RSA. A bid is a key pair.	Passive	Computational
[68]	Hash functions and digital signatures	Passive	Computational
[38]	Commitment Scheme	Passive	Unconditional

2.2 VICKREY AUCTIONS

In a sealed second price auction, bidders submit to a bid taker a sealed value presumably of one's true valuation. Bidders can submit as long as it is not past the closing

time. Once the allowed time of submission has passed, no more bids are accepted. Entering the opening phase, the winner is defined as the entity with the highest bid, and as the rule dictates, the winner pays the second highest valid bid. In one of the earliest publications, in 1993, the authors of [40] proposed the idea of constructing a second price sealed-bid auction using cryptographic protocols. During the opening phase the protocol was concern with hiding the values of all submitted bids. Once a winner was determined, only the second largest would be revealed in order to know the amount to be paid. The authors suggested a series of six steps to be followed in their protocol. The protocol suggested by the authors is describe for the case with only two bidders A and B, plus an auctioneer C. Bidders can represent a bid by a value on the interval [1,100]. Both of the bidders proceed to submit an encrypted message that is encrypted with the auctioneer's public key plus their own public key. One of the bidders calculates the difference between the encrypted number, labeled k, and the ordinal value labeled j. The receiver calculates a sequence based on the equation y_u = dA (k-j + u), where u is all possible values [1,100], and also computes $z_u = y_u$(mod q) where q is an arbitrary prime number. Then the sequence is sent back, from which the other bidder can determine if his value is strictly larger or perhaps smaller.

In another scheme, and one of the most widely cited papers in electronic auction protocols is [19]. One of the key reasons for the success in the protocol is the use of verifiable signature sharing [18], and for being one of the first to formally introduce a secure method for second-price sealed-bid auctions. To begin with, the auction utilizes the primitives of group multicast, secret sharing (t,n) threshold scheme [65], digital cash, and as mentioned verifiable signature sharing. The service is constructed using more than one auction server, and requires a third of the auctioneers to collude in order to harm the auction. Essentially bidders submit bids to the corresponding server by splitting a digital coin in the form of (v$, Obank(v$), ws), using (t,n)-

threshold secret sharing, except for the middle item which uses a verifiable signature sharing primitive. When the bidding period has ended, auctioneers reconstruct the bid values using one of the suggested group multicast primitive, and declare a winner after comparing results. Finally, the house can collect the money easily since the verifiable signature provided is giving such right of ownership.

Under the categories of sealed-bid auctions implementations without threshold trust is [35]. The protocol requires the use of an oblivious third party, labeled as the auction issuer which is particularly in charge of constructing the circuit to be used by auctioneers. The protocol behaves with a property of fairness mainly by avoiding leakage of information even if the auction issuer and the auctioneer collude. Bidders must communicate directly with two servers. Bids are encrypted before sending to the circuit. Using multi-party computation, realized as a boolean circuit, decryption keys are shared among a few of the auctioneers. The auctioneer publishes the result to be publicly verified.The protocol efficiency is improved by [32] using a homorphic scheme instead. The limitations of one of the servers cheating is the topic of [26], where the improvement is to split the bid into two shares that under field sum represent the original share.

An additional scheme [4] proposes a protocol which resolves one of the major issues of Vickrey auctions, more specifically, the problem that arises when the dealer lies about the actual value of the winning price to make a profit from the players. The proposed protocol can is built with two major phases to the auction. In the first phase, each bidders encrypts a binary bid list. Encryption occurs over every bid with a different unique personal key. Encrypted bids are published onto a black board as a bid matrix. By publishing each bid onto a black board anonymously, the auctioneer cannot insert fake bids. In the next step decryption over the bids by the public keys determines the winner. In order to efficiently find the winner, and to not reveal any unnecessary information, three methods are offered by the author. The three search

methods include: downward search, upward search, and binary search. Once a winner is found, the winning key is published. Since this is unique, only the winner can prove to be the winner of the auction. Finally, the protocol enforces a fine on players that attempt a key denial attack.

In a later publication [5] also by Felix Brandt a protocol called YMB-Share that can be realized without auctioneers was proposed. In this protocol, bidders once again create binary entries for choice of price. There is yes or no choice for the price. Bidders submit shares of their bids and must compute a jointly function on all shares received for each discrete price. A personal key is associated with each bidder and price. All calculations are performed in a finite Albelian Group. For instance, computing keys is performed using ring transfer. Ring transfer is also used to determine the winning key. One important aspect is that bidders can only jointly find out the winning key, and nothing else is revealed. Once the winning key is published, the bidder holding the same key can be determined as the winner. Public verifiability can be accomplished by requiring the bidder to authenticate by supplying a signed message containing the key.

Omote and Miyaji [43]created a protocol with verifiable discriminant of p_0-th root which requires no anonymous channel. The protocol requires two auction managers who are in charge of different tasks. AM1 handles bidder registration, and AM2 manages the bidding phase. During the process of bidding, both AM1 can verify the validity of the bids, while during opening any one can verify validity. The protocol is constructed with signatures based on proof of knowledge, public key encryption ELGammal, and verifiable decryption mix. AM1 chooses values $t^0_{i,k}$ and $t^1_{i,k}$ that have the $p_0 - th$ root. Bidders have the liberty to see all possible prices from AM2 database. After selection choice, bidders, send a public key with signature which depends on values $t^0_{i,k}$ and $t^1_{i,k}$ chosen by AM1. Validity over the bids is checked by using decryption mix, and its a matter of showing the bids have the $p_0 - th$ root. During the

opening phase, computation of $M(X_k)andM(Y_k)$ informs the auction managers if a bid was place that was higher than k. For example, if $M(X_k)andM(Y_k)$ return (1,0) for the point k, than no one submitted higher than this point. If $M(X_k)andM(Y_k)$ returns (0,1) , than at least one of the bidders place a bid at that price. A last possible scenario is that $M(X_k)andM(Y_k)$ returns (0,0), meaning multiple bidders placed a bid at this value or higher than this value. The problem is that cases (0,1) and (1,0) are indistinguishable. Once a winning bid is decided, then a winner is found by AM1 and AM2 working together.

Aside from using cryptography methods in the auction protocol, [17] interleaves cryptography in the auction mechanism, essentially creating an auction mechanism unlike the classical auction mechanism. The protocol consists of R number rounds, each being a second price sealed-bid auction, however, the essence of the protocol is a mechanism created with parameters ($\epsilon and m$) that create a trade-off and allow for flexibility between important desired properties such as resource-effectiveness, cognitive cost, security and privacy. Bidders must obtain commitment keys, encryption keys, and signature keys. A monotonic bijective function from the possible valuations to the actual valuations is used by bidders to create their bids at each round, that is $b_i^r = \phi(B_i(e_i^r))$. Precomputed values are allowed at each round of the protocol to reduce round complexity, and computational cost. Every bidder submits an encryption of b_i^r and argues that $\phi(\frac{1}{1-\epsilon}\phi^{-1}(b_i^1)) \geqslant b_i^r$ in zero-knowledge proof. The auctions is said to finish once the winning price of the current round and its corresponding bidder, is the same from the previous round. If a tie occurs, the ultimate winner is chosen by the equal probability rule.

Mehrdad and Stinson [39] constructed two secure second price sealed-bid auction using masking techniques and verifiable secret sharing [66], although the protocol can also be constructed with the more complicated VSS of [58]. In the first implementation bidders hide the bid by masking it using + operation of two shared secrets. In the

second implementation the bidders hide the bids using both the + and x operations of two shared secrets.

2.3 M+1 AUCTIONS

An M(+1) auction is a form of auction in which the M highest bidders take the prize, and the (M+1)-st is the amount to be paid. In the case of M=1, the protocol resembles a Vickrey auction in which the second highest price is paid. Papers such as [27] [1] [6] [7] [70] contain details on how to set up a secure sealed-bid auction.

The idea behind [27] is an multi-party computation scheme. Instead of hiding the secret in the sum and products of the free variables, the protocol hides the secret in the degree of the polynomial. The approach for a secure auction protocol is to compute the product of the secrets in a way that the resulting polynomial returns the number of bidders willing to pay at a specific price. The auctioneers determine the winner by polling until the highest price is found. After identifying a winner, he/she must prove to be a true recipient of the prize. In order to realize an (M+1) auction the auctioneer remove the winners from the set, and reiterate the process to the find the next set of winners.

The approach presented in [1], consists of bidders, auctioneers, and trusted party. The trusted party generates the public key and private key in preparation for ElGammal public key encryption. Bidders use their available keys to generate publicly encrypted bid vector. They also compute the differential of the bid vector. Auctioneers take the integral to recover the information along with a mix and match procedure to test if a bid is lower or higher than a predetermined value. The search for the winner is performed via a homomorphic binary search.

Felix Brandt published two fully private (i.e no auctioneers or TTP are used to solve the auction) protocols for (M+1) auctions.The publication [6] improves some of the issues in [5] related to leaking information when bids are equal, lack of verifiability, and the need for bidders to have to share exponential shares. Similar to other protocols already mentioned, there is an ordered set of k possible prices $p_1, p_2, ...p_k$. Each bidders sets a differential bid vector, and distributes it on all other bidders. Finding the highest price requires shifting down the components of the bid vector. Each bidder b_i contributes a final computation, which is multiplication on the shares. If the multiplication changes value 0 to 1, then that bidder b_i producing the 1 is the winner. Public verification is achieved inherently as a result of using VSS in the protocol sharing phase.

Another protocol for (M + 1) auctions is proposed by same author in [7] using ElGammal crypto system. Bidders jointly compute the auction protocol results in constant rounds, usually 3, regardless of number of bidders, and combination binds.

A construction of an M+1 auction is used under a different context in [70]. The focus in this protocol is shifted from price being the unique strategy dimension, to focusing on the quality offered by an item, or the attributes of a deal. In this context, auctioneers are buyers, and bidders are sellers, that is, there is a single buyer (i.e. government), and a set of sellers $N = \{1, 2, ...n\}$. Each item has a cost and associated quality, $c(\theta, q)$. The gross quality of a buyer is $V(q)$, and the payment to the i^{th} seller is p_i, therefore the utility of a seller is $p_i - c(\theta, q)$, and that of the buyer is $V(q) - p_i$. In the first step, bidders send (q_i, b_i) in an encrypted format applying homomorphic encryption. A second prize winner is found using the same technique as [1]. After the second prize winner is found a decryption of the quality $D(E(q_i))$, and calculation of $V(q_i) - b_{2nd}$ assigns the final payment.

2.4 RULE FLEXIBLE SEALED-BID AUCTIONS

Previously, it was established that an auction in which the winner pays his own price is regarded as first price while paying the second highest is referred to as second price. Some of the protocols mentioned above have been designed for a specific setting, namely, first or second price auctions. The next set of protocols are simply concern with providing a method for sealed biding in the more general sense with a flexibility to the rules of the game. In essence, the next papers [67] [23] [52] [41] [34] and [49] can be applied to a first price or second price auction environment.

In one of the first papers on the topic [23], a protocol is design to utilize the idea of distributed computation based on the work of Ben-Or, Goldwasser, and Wigderson [2]. More generally, the authors describe a protocol that could resolve ties, and would never reveal bids to any party, even after the auction completed. In essence, the protocol distributes information among auctioneers by means of polynomials providing t-privacy, and t-resilience properties. From the two operations, multiplication and addition, the most significant operation used throughout the protocol is the addition of polynomials based on the fact that the value of a sum of polynomials is the sum of the values of each polynomial evaluated at the point, and that polynomial multiplication encounters several difficulties among rapid increase of polynomial degree, and the possibility of choosing an irreducible polynomial during the protocol. During the bid submission phase, bidders encode their bid, which is a value from a price list. Each of the digits of the bid is encoded by a secret polynomial. The bidders proceed to distribute their shares to auctioneers. Later auctioneers perform multi-round computation (one for each digit) upon the encoded information to find the largest or second largest bid i.e. the selling price. As a final stage the bidders bids are summed by the auctioneers working together to find the id of the winner. If a tie occurs, no one single bidder will be able to claim the winning Id. In the situation of a tie, a second round with a new price list takes place for the participants that tied. The process continues

iteratively until the protocol can distinguish a single winner. Moreover, the protocol can be improved by [29], providing an improvement on privacy at double the cost of computation. Lastly, the proposed protocol can be improve as suggested by [54] to provide fairness, anonymity, and robustness.

In the category of auction design with public cryptosystem, Subramanian [67] designed a 6 step protocol. The protocol relies on private and public key encryption as well as a broadcast channel. The protocol had several deficiencies on efficiency and data manipulation by malware or malicious users. In order improve the protocol [24] introduced the notion of timestamp into the protocol. However, while the timestamp provided with an extra challenge for a malicious user, it did not completely prevent data from falsification. By introducing a one time registration stage, [31] improved the protocol by Hwang. One of the drawbacks left unresolved in Liaw's protocol was a third party conspiracy with a bidder, which was remedy by [72]. The authors of [52] innovated a protocol that uses mix networks as a main tool. The set of players generate one way collision resistant hash functions on some input $c_i = H(b_i v_i)$, where $b_i \in \mathbb{Z}_q$ a mapping from $y \in \mathbb{Z}_q$ public keys. Applying either decryption chaining [14] or re-encrytption [45], all the c_i are shuffled, and later in another round (b_i,v_i) are shuffled in the mix network. Proceeding shuffling, the permutations and commitments are posted on a bulletin board by a set of servers. During the opening phase decryption among all bids occurs. In order to determine a winner, a player must prove commitment matches that of the winning bid.

Authors Baudron, and Jacques [41], create a homomorphic protocol using boolean AND and OR gates. In the initial step of the protocol, participating players must publish their public keys. Each of the participants encrypt the price of choice under all other public keys including his own. A boolean circuit is introduced in the protocol as part of function that determines the highest or second highest bidder. If the protocol requires protection against malicious adversary, a similar protocol is

deployed but using Paillier's encryption scheme.

Moreover, auction protocols can be realized with an efficient MPC. Normally, MPC with verifiable secret sharing (VSS) for robustness, requires bidders to submit ciphertexts and zero proof knowledge, which result in a larger overhead. However, one approach to improve MPC, and avoid VSS altogether is to create a private decryption key that is shared among the servers. These idea is describe in the works of [15] [25]. Both forms of efficient MPC, that is mix and match method, and additive homomorphic cryptosystem are extended in the work of [34] to create an efficient sealed-bid auction where a bidder submits only one ciphertext and experiences only a few multi-exponentiations.

In a homomorphic scheme, the authors of [49] create a protocol based on a modify version of Goldwasser-Micali Encryption [21]. In the set up, a broadcast communication channel is used by m auctioneers $A_1, A_2, ...A_m$ to provide the different G-M encryption scheme, and public key. Bidders have to select from a discrete set of price, and represent their choice as a bid vector with -1 indicating participation at a specific price, while simply 1 indicates opting out. The auctioneers perform a binary search to find the winning price. During the binary search, if a decryption returns positive, the search head upwards, otherwise it takes a downward direction. Once a winner is found, in order to public verify, and proof correctness, a ZK proof is implemented. For ciphertext returning -1, a proof of knowledge of square root of ciphertext suffice. On the other hand if ciphertext returns 1, a proof of knowledge of a square root of cy will suffice. The ZK proof is based on earlier work of [22]

2.5 COMBINATORIAL AUCTIONS

In contrast to the above mentioned auctions, combinatorial auctions allow players to bid on any number of combination of items called bundle or set. Combinatorial auc-

tions can be multi-unit (multiple units of the same item), linear-good, and general auction (a set of different items). Linear-good auctions consist of a set of sequentially ordered goods G, and bidders tend to bid such that they obtain a sequence of good[71]. Cases where combinatorial auctions are in use include the sale of furniture, spectrum auctions held by Federal Communication Commission [33], and the sale of airport time slot. An inherent challenge of combinatorial auctions, referred to as combinatorial auction problem (CAP), or winner determination problem, is that of computing the optimal solution, that is, that set of disjoint goods such that the sum of the goods is maximize. Papers such as [20] [59] [62] [63] have provided possible solutions to the winner determination problem. Most of the time, for secure auctions the solution is solved with a dynamic programming approach since it is a strong tool for solving longest or shortest path approach on a directed graph. In general, the idea behind combinatorial auctions is to make a profit from selling desirable complementary items, while finding the winning party is a matter of solving a hard combinatorial optimization problem, usually implemented with a dynamic algorithm. An in depth coverage of combinatorial auctions is presented in [16]. Secure protocols for combinatorial auctions are part of the publications of [69], [73], and [46].

Taking an approach of dynamic programming and using Shamir's secret sharing in a similar manner as [27] (the secret is related to the degree of polynomial instead of the constant term), the authors of [69] have constructed a secure combinatorial auction for the general case and effective against the passive adversary. The method employed consist of using secret sharing to share secret among publisher and evaluator of the weight of a link. Evaluators come together to solve the optimal value using secure dynamic programming. In a final step evaluators trace back the links to obtain the optimal solution, or the longest path. Due to its nature this type of construction generates a protocol with unconditional security.

The next protocol [73] utilizes homomorphic public key encryption of ElGammal to

realize secure dynamic programming. Bidders are represented by weight publishers. Part of the auction servers are represented by evaluator. Evaluators knows only its own valuations. At the start of the protocol, the weights are encrypted with public key encryption. Because ElGammal provides indistinguishable, homomorphic, and randomizable, the wights are each encrypted with different random value r, and a random constant f is added. At winner determination, the optimal value is found by decrypting the j-th element from the componentwise product and checking if the value is equal to 1. When the value found is not equal to 1, then it can be concluded that a maximum has been found.

As mentioned, the paper [46] provides another solution to combinatorial auctions. The approach taken by the author is to use Paillers'e encryption as a cryptographic primitive. At the start of the auction, the auctioneer provides with a public time-lapse cryptographic key N [57]. Auctioneer proceeds to publish price vector, followed by bidders encrypting and submitting bids. During the winner determination phase, a branch-and-bound algorithm is used to solve the optimization problem in the plain text bid. The plain text is used for reducing the time complexity, however, even though the bids are revealed, the auctioneer cannot modify or change the outcome of the auction.

In addition Mehrdad and Stinson [39] created two protocols for combinatorial auctions. The protocols require using Initializing party, the + operator and and max functions in order to perform secure sealed-bid auction. Furthermore, the auction models are represented as a case of multiple traveling salesman problem and are solve using dynamic programming techniques in one implementation, and inter-agent negotiation in the second implementation.

Table 2.2: Vickrey Auctions Summary

Reference	Cryptographic Method	Adversary Model	Security Model
[40]	Uses public key encryption, compares two bidder at a time	Passive	Computational
[19]	First to bring Verifiable Signature Sharing	Passive	Computational
[35]	MPC is used to share Public/Private keys	Active	Computational
[32]	.Homomorphic Encryption	Active	Computational
[4]	Uses public key cryptography, and three possible ways of searching: downward, upward, and binary	Passive	Computational
[5]	Differential Bid Vector. Shifting down of vector along with user input retrieves the highest price.	Active	Computational
[39]	Verifiable Secret Sharing and masking using + and x operations	Active	Computational
[43]	Zero knowledge proof signatures, and ElGamal cryptosystem.	Active	Computational

Table 2.3: (M+1) Auctions Summary

Reference	Cryptographic Method	Adversary Model	Security Model
[27]	MPC. Bid value is in the degree of polynomial	Active	Unconditional
[1]	A TTP generates key pairs that are used in a mix and match approach	Passive	Computational
[6]	Bidders create a differential bid vector. Bidders share with other bidders. A bidder computed number determines the winner..	Active	Unconditional
[7]	A sealed-bid auction built around ElGamal	Active	Computational
[70]	Auctioneers are buyers, and bidders are sellers. Bid are encrypted using homomorphic encryption	Passive	Computational

Table 2.4: Rule Flexible Auctions

Reference	Cryptographic Method	Adversary Model	Security Model
[67]	Encode each digit of bid by a polynomial. MPC over every digit produces the winner.	Passive	Computational
[23]	Uses public key encryption as well as a broadcast channel	Passive	Unconditional
[53]	An auction with mix networks	Passive	Computational
[41]	Homomorphic encryption is to encrypt bids. Boolean circuit determines highest bidder.	Passive	Computational
[49]	Create a private decryption key that is shared among servers, to avoid VSS. Uses homomorphic encryption to avoid communication complexity of MPC.	Passive	Computational
[34]	An auction based on Goldwasser-Micali Encryption	Passive	Computational

Table 2.5: Cobinatorial Auctions

Reference	**Cryptographic Method**	**Adversary Model**	**Security Model**
[69]	Combines secret sharing and dynamic programming.	Passive	Unconditional
[73]	Homomorphic encryption based on ElGamal to realize a secure dynamic programming solution	Passive	Computational
[46]	Uses Paillers's encryptions, and time-lapse cryptographic key. A branch and bound algorithm is used for optimization.	Passive	Computational

CHAPTER 3
COMMUNICATION AND COMPUTATION ANALYSIS OF SEALED-BID AUCTIONS

Although many works in the literature discuss theoretical approaches to sealed-bid auctions, the research on implementation is limited. Herein, our aim is to elucidate the intricate details of implementation, and provide computational, and communication analysis of several auctions.

The protocols that were considered are: [27], [60], [61]. For [27] two version were implemented, with and without verification. The verification version prevents bidders and auctioneers from misbehaving. For [60] there is also two implementations, one for cryptosystem ElGammal, and another one using RSA. Finally we conclude with an implementation of [61].

An in depth assessment of each auction is generated by modifying parameters such modulus size, number of bidders, and price range. Results show complexity increase with modulus sizes as expected for all auctions. Interestingly, for [27] complexity increase more with increases in price ranges. Finally, the work with highest communication complexity when number of bidders is increased is [61].

3.1 PROTOCOL DESCRIPTION

So far we have covered some important works in the literature. We have also mentioned desired properties, and security models. Next we explain in detail the selected protocols that are the subject of our study.

3.1.1 Hiroaki Kikuchi

The proposed work is based on the addition properties of multiparty computation [2]. The protocol relies on the important proof that if f and h are polynomials of degree t and s respectively, then f + h has degree max(t,s). First, a prime p of order q is chosen. Auctioneers publish a price list W, and the item for sale. Each bidder chooses a random polynomial of the field and the bidding value is equal to the degree of the polynomial. Each auctioneer receives a share from a bidder, and compute a total sum denoted as F over the shares receive. Because polynomials were chosen so that the free variable was equal to zero, any entity can find the smallest subset that interpolated using Lagrange will produce a polynomial containing the highest bid as the degree of the polynomial.

The second version of the protocol is made stronger with information theoretic verifiable sharing [48]. First, a prime p of order q is chosen same way as before, but now also two distinct primitive roots g_1 and g_2 are chosen and made publicly available. Bidder b_i chooses two random randomly generated polynomials f and h. Each bidder makes a commitment of the polynomial by sending the multiplication of the powers of the primitive roots with the coefficients of the polynomials. For example, for polynomial $f(x) = a_1x + a_2x^2 + ... + a_t^t$ and $h(x) = b_1x + b_2x^2 + ... + b_s^s$ the bidders sends $g_1^a g_2^b 1$, $g_1^a g_2^b 2$, until a commitment is made on all the coefficients. The sums F and H are calculated for the shares received from f and h respectively. Auctioneers calculate $Y = g_1^F$ and $Z = g_2^H$ and publish YZ. Interpolation once again generate a polynomial that contains the highest bid in the degree. The main difference is that with the extra computation to incorporate the verification protocol, neither auctioneer or bidders are permitted to insert fake values. Committing at every step of communication makes cheating trivially detectable.

3.1.2 Kazue Sako

Two practical cryptosystems, ElGamal and RSA, are used in the computationally secured sealed-bid auction [60]. The design centers on the idea of group decryption. For each price in the price list there is an associated public key. For example, for price list $V = \{v_1, v_2, ..., v_L\}$ there are public keys $PubK = \{pubk_1, pubk_2, ..., pubk_l\}$, and the auctioneers keep in their servers private keys $Pk = \{pk_1, pk_2, ..., pk_l\}$. In order to bid, a bidders uses the key associated with a price, and encrypt the bid with that key. During the winner determination phase, the auctioneers pick the private key associated with the highest price on the list and try to decrypt every submitted bid. A successful decryption indicates a winner, if no winner is found at a specific price, the auctioneers pick the next highest private key, and reiterate the process. In order to make the protocol stronger, the authors suggest using [65] to split the keys into n shares and give a share to each auctioneer to provide a mechanism of resilience against dishonest auctioneers wishing to decrypt all bids.

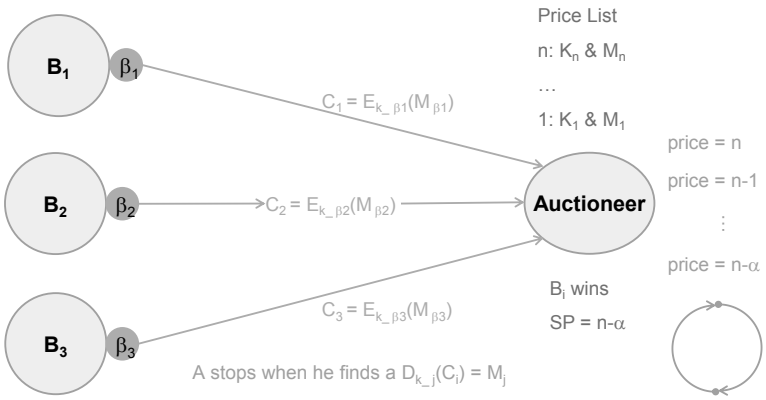


Figure 3.1: Kazue Sako Auction Model

3.1.3 Sakurai and Miyazaki

The publication [61] describes an auction that can be built using a convertible undeniable signature scheme [3]. First, the system needs a safe prime p, a subgroup generator α, and an one way hash function. The prime p and the primitive root α are used to create secret that are computationally secure based on the discrete logarithmic problem. The auction proceeds in a Dutch style fashion. The verifier (auctioneer) and the prover (bidder) engage in several rounds of communication to prove equality or inequality against the standing price. Determination of equality or inequality does not reveal any private information since the auctioneer is arriving at a conclusion based on comparison of two discrete logs. At the point of finding a winner, bidders reveal their private key which further confirms they are the winner.

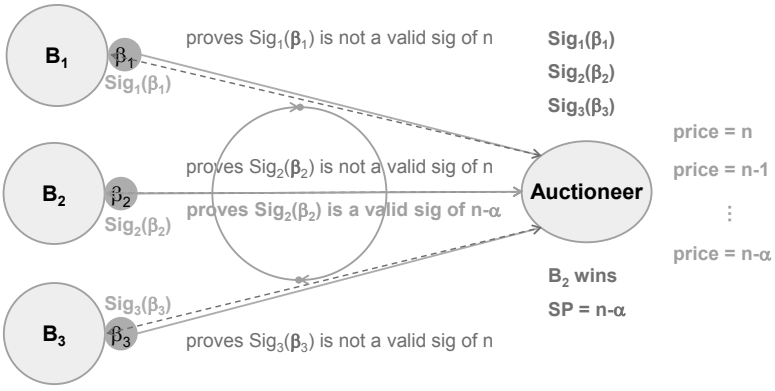


Figure 3.2: Sakurai and Miyazaki Auction Model

3.2 RESULTS AND DISCUSSION

Our simulation was developed under JetBrains CLion environment. Our implementations were written in C++ and compiled under GNU GCC compiler. For rapid development we relied on Crypto++ library for cryptographic modules. For instance, we made extensive use of hashing algorithm SHA1, and used public crypto systems such as RSA, and ElGamal. In addition, Crypto++ provided with the necessary blocks for generating random safe primes, prime numbers, and primitive roots of a cyclic group. In order to generate random polynomials we applied random number generators for each coefficient, generating n-elements and applied modular reduction to keep elements in the finite field Z_p. Polynomials were stored as a vector where the first element mapped to the constant term and the last element mapped to the leading coefficient.

Our experiments were centered on computational complexity and communication overhead around initialization and verification. For each protocol we tested initialization based on four distinct modulus bit size: 128, 256, 512, 1024. Evaluating verification was measured by allowing bidders to have different bidding preferences, and by manipulating the number of bidders present during the auction. For instance, we measured verification complexity in scenarios containing 25, 50, 75, and up to 100 bidders. At the same time bidders could choose a bid b_i at random from the whole set if the bidding parameter was 100%, otherwise they would have to choose only from a subset. For example, if the set contained 100 elements in numerical order, then bidding parameter 75% essentially meant the bidders would ignore bidding the top 25 elements, and instead would choose a bid b_i at random from the remaining 75 lower order elements. Hereafter, we will refer to this bidding preference, which we tested for at 25%, 50%, 75%, and 100%, as price range parameter.

For simplicity, the experiments were created using a Command Line Interface and a common graphical user interface. Our computation model was at all times

synchronous, thus it introduced some delay. However, the delays introduce by this model was relative. Some areas that the were affected by synchronous model was creation of public/private key pairs, generation of polynomials, and publishing the results to our simulated bulletin board.

In terms of our computing power, we conducted our investigation in a computer with Intel Core i74810MQ CPU @ 2.80 GHz and 16GB RAM. While running the experiment we ended every process that would steal computing power from the operating system. In addition we shut down the network, and closed down any ports. To the best we could we made the operating system solely focus on running the experiment.

3.2.1 Multiparty Computation

The first set of results correspond to non-veriafable protocol described in [27]. Naturally the initialization increased with an increase in the modulus size. The first reason for an increase is that generating a random prime number becomes increasingly computationally expensive with greater bit size, and at best the algorithms produce only probabilistic prime numbers. The second reason for the observed increase is that for each polynomial we chose random numbers based on the size of the field, thus modular reduction increased as well as the numbers became larger.

The verification graphs show time for different number of bidders, and varying price ranges. The verification seemed to be more affected by price range, and bidder size rather than bit size of the modulus. Two factors contributed to the increased in time observed. First, in this protocol, the private value of the bidder is concealed in the degree polynomial, this means that in order to conceal a bidding value of 100$ we need to construct a polynomial of degree 100, meaning we have more computation to manage when we are evaluating shares, and using the addition property of MPC.

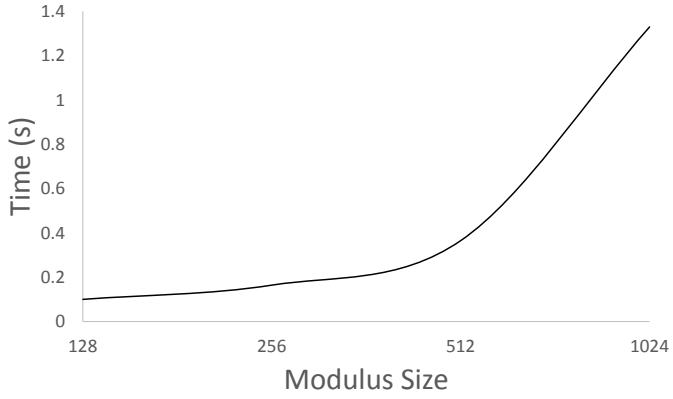


Figure 3.3: Initialization Time (Kikuchi MPC)

Secondly, with increasing price, we need to increase number of auctioneers if we want the same security threshold. However, increasing number of auctioneers increases communication complexity during the sharing, and reconstruction part.

Figure 3.1 shows the initialization time for the same protocol when verifiability is introduced as a method to prevent bidders from repudiating, and auctioneers from casting false bids with the purpose of inflating prices. The result of adding extra layers of robustness is an increase in computation. Unlike in the simpler approach, we must generate a prime and two distinct generator of the cyclic group. Also bidders must generate two distinct polynomials f(x) and h(x). Also, for each coefficient in the polynomial the bidder must submit commitments. Then the auctioneers must verify the submitted commitments before they can establish a trust in the bidder. The net result is an increase in time, due not only to higher modulus size but also the extra number of precautions added to the protocol.

The same pattern as the simple implementation can be observed in the verifiable protocol, that is, with greater number of bidders, and greater number of price ranges, we note an increase in time. The extra accumulated time, in comparison to the simple implementation, is due to communication complexity and extra verification

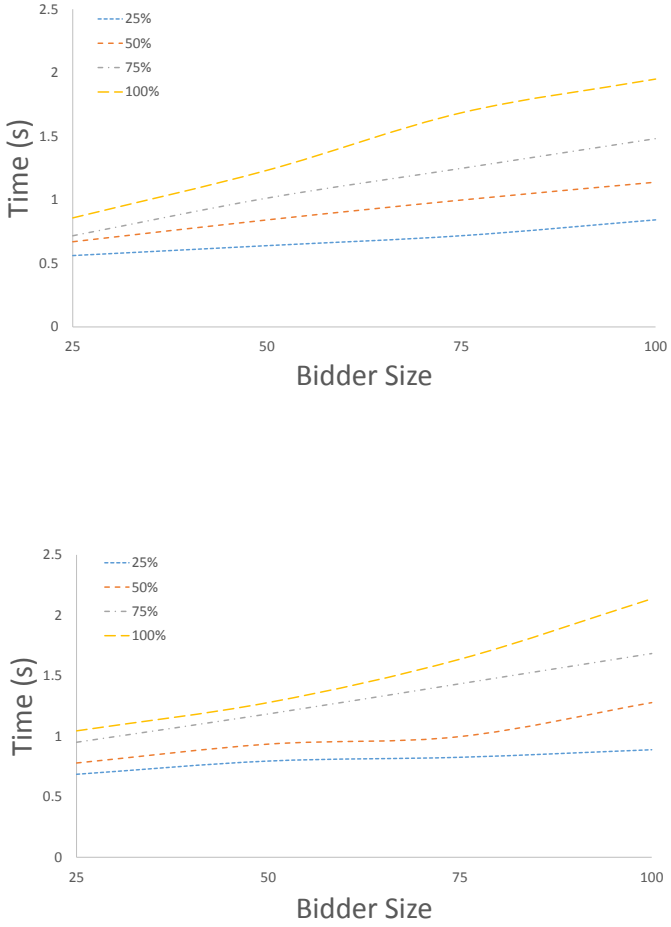


Figure 3.4: Verification Kikuchi MPC: 128 bits & 512 bits

steps. Auctioneers must publish two constants Y_j and Z_j based on the multiplication computed from $g_1^F g^H$ for each corresponding α_j. As another additional step, each concerned entity, mainly bidders, must afterwards verify that the auctioneers computed their values correctly. Finally, reconstruction is similar to before.

3.2.2 Public Key Encryption

Below we have plotted the initialization time for [60]. The first observed set of plots figure 3.5 and 3.7 corresponds to implementation utilizing ElGamal cryptosystem,

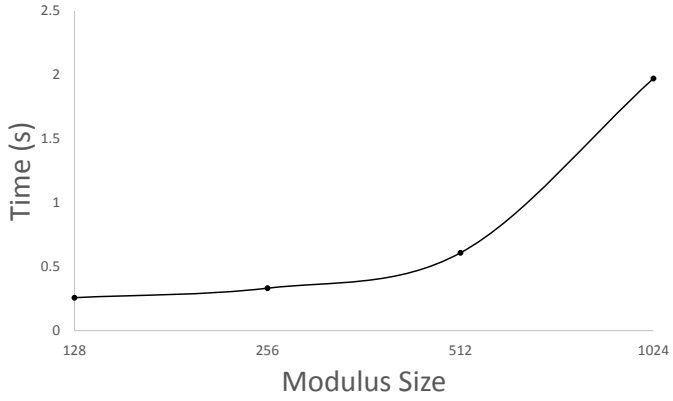
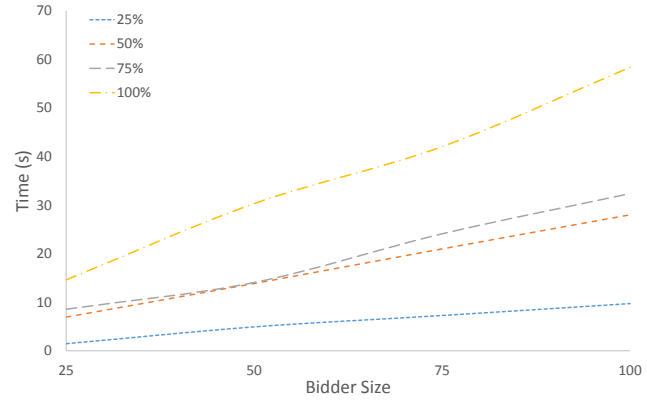


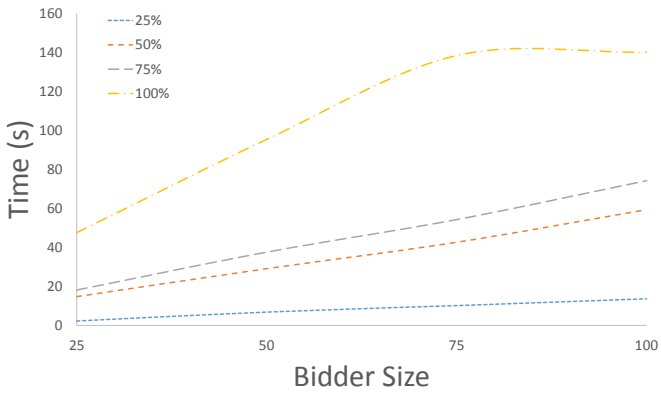
Figure 3.5: Initialization (Kikuchi Veriable)

while the next figures 3.6 and 3.8 relate to RSA. In both Initialization time rises with modulus sizes as it is increasingly hard to generate large primes. The time for ElGamal is higher than RSA since the prime generation algorithm needs to find prime p for which (p-1)/2 is a safe prime again, while RSA only requires to large unrelated primes. Clearly the number of price ranges also affects the total initialization time since for each element in the set a decryption/encryption key pair must be generated.

One expected outcome, is that increasing the modulus size increases the computation complexity since now we have bigger bit size keys for encryption and decryption. More subtle, however, is the fact that we see an increase in verification time as the number of bidders participating increase, while producing an inverse relationship for the price range. We can justify the observed behavior of the price because of the Dutch style nature of the auction. When we set the price range to 100% bidders an we have 100 bidders participating, it is very likely that one of the bidders will bid the highest price. Since decryption occurs from highest decryption key to lowest, essentially we will greatly reduce communication expenses if we find a bidder that bid equal to or very close to the highest price. On the other hand, when the price range



(a) 128 bits



(b) 512 bits

Figure 3.6: Verification time (Kikuchi Veriable)

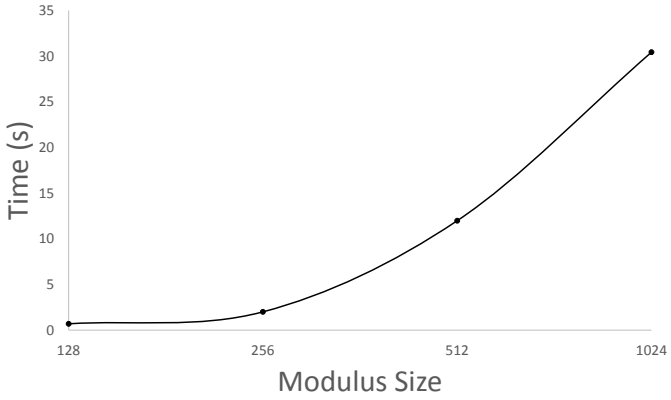


Figure 3.7: Initialization (Sako ElGamal)

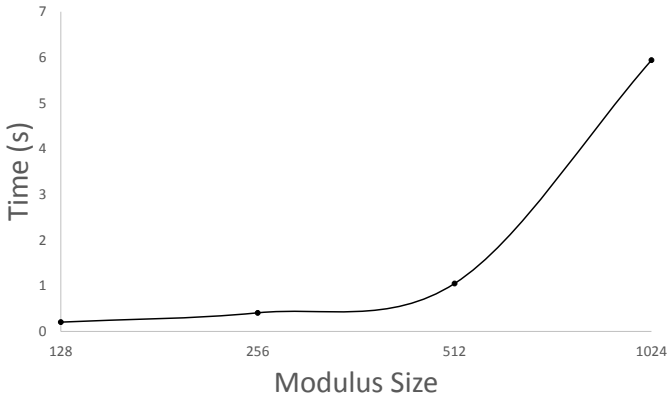
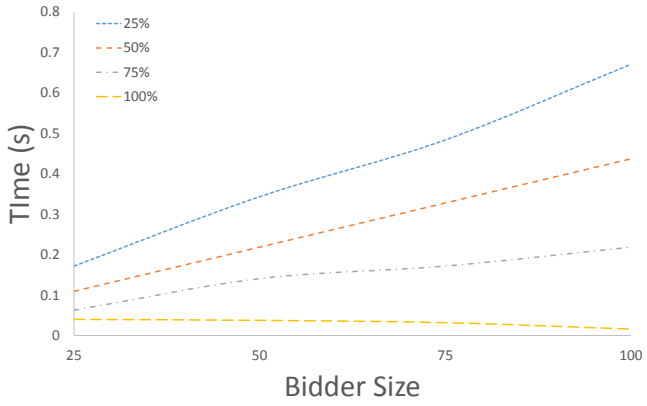
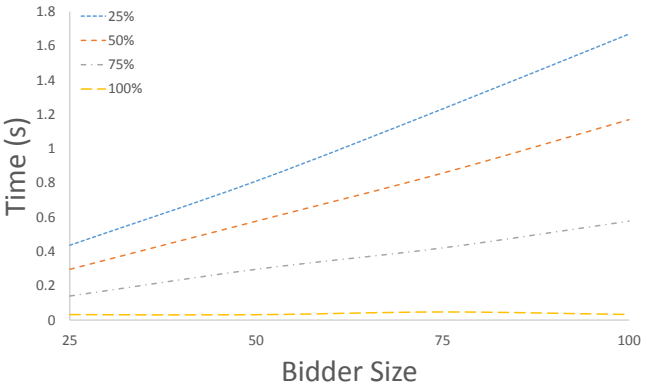


Figure 3.8: Initialization (Sako RSA)



(a) 128 Bits



(b) 512 Bits

Figure 3.9: Verifications Sako ElGamal

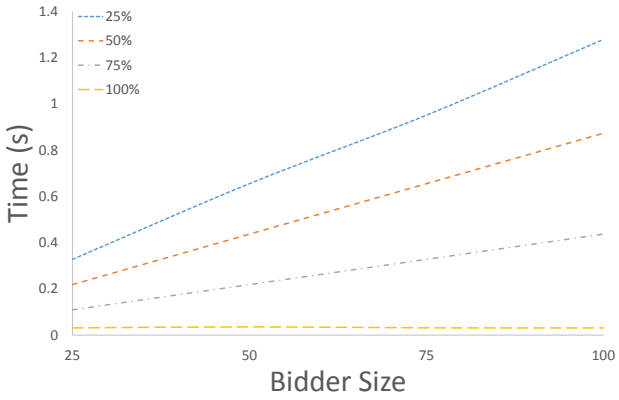
parameter is set to 25%, the auctioneers waste a vast number time decrypting values for which nobody placed a bid. Therefore, for 100 bidders, and 100% the result is minimized, while it is maximized at 25% and 100 bidders.

Overall the protocol proposed by Sako can be implemented by any public key cryptographic system. Our choice for ElGamal, and RSA is due to the fact that these two are well known cryptographic systems. In the end, the verification times were very similar, with RSA being a bit slower than ElGamal. Although encryption is faster for ElGamal, RSA has significant advantage during the initialization steps because of slow prime generation present in ElGamal.

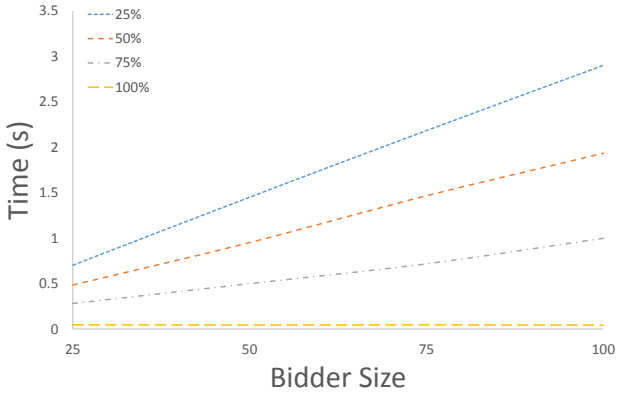
3.2.3 Commitment Scheme

In this section we examine the protocol propose in [61]. The first step in the design of the protocol is to generate two primes (p, q) with the condition that $p = 2q + 1$, and α, a generator of the subgroup of Z_p of order q. Following that, bidders must compute private and public values. Because no encryption, or decryption is needed during initialization, this protocol is the fastest to initialize. The complexity increase seen with increasing modulus size is solely due to the computation time needed to generate a safe prime number, and a generator.

During the bidding process, bidders commit to a bid, and attach a digital signature. In the opening phases, auctioneers must start at the highest possible price and receive a proof from each bidder showing equality or inequality in dutch style. Therefore, the verification time needed for the protocol is proportional to the number of bidders, and inversely proportional to the price range. The plots are minimized at 25% price range, and maximized at 100% price range. It appears that the modulus size, significantly impacts verification. To understand the result we simply analyze one key important step of this protocol, specifically when auctioneers compute $\beta = \alpha^s P_j^{H_i(w_m,r)}(modp)$. Essentially we need to perform hashing, and then we have



(a) 128 Bits



(b) 512 Bits

Figure 3.10: Verifications for Sako RSA

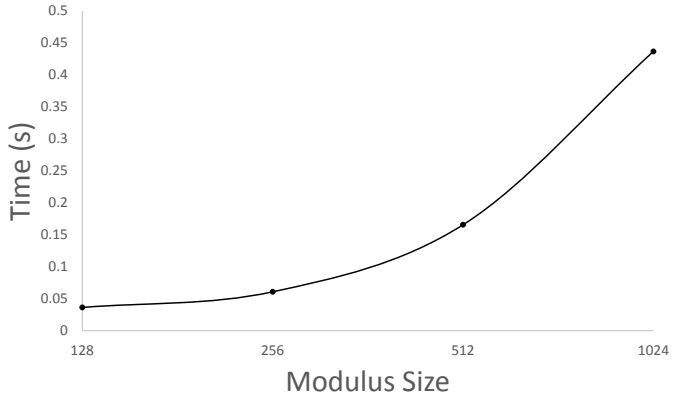


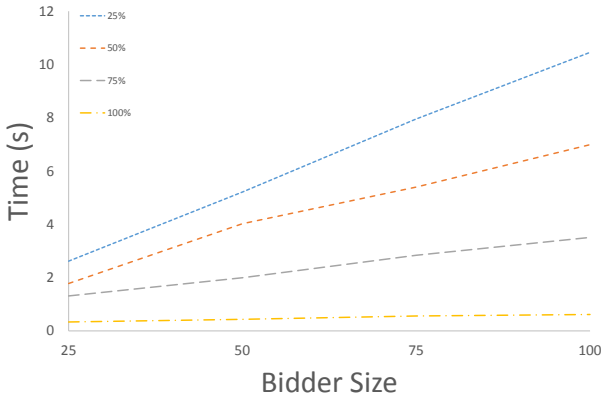
Figure 3.11: Initialization (Sakurai & Miyazaki)

exponentiation of large numbers. Afterward, the auctioneer and the bidder engage in the protocol describe by Mitchels-Stadler for proving equality/inequality of two discrete logs. Since the auction is constructed in a Dutch style manner, auctioneers and bidders must exchange information over several rounds before finding out a winner. Once a bidder successfully proves the equality of his bid with the current standing price, an extra step is required to prove confirmation. In the confirmation step the bidder must send the auctioneer his private exponent x in the discrete log, and finally the auctioneer can confirm if the supply parameter satisfies $r = (\alpha^s P_j^{H_i(w_m,r)})^x (mod p)$.

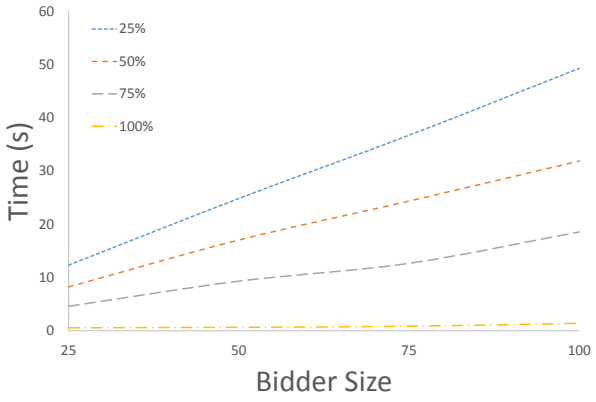
3.2.4 MPC vs Public Key Encryption vs Commitment Scheme

In figure 3.11 and 3.12 we provide a unified comparison of all the protocols. In order to fit all the result in one visible plot we applied a logarithmic scale for the time axis. For initialization we considered 128, 256, 512, and 1024 bits. For the verification we have plotted 512 bits, 50 bidders, and price range 50%.

Within initialization plot we see that the Sako public key ElGamal protocol is the most computationally expensive. It should be noted that the slow initialization is not that important since preparation of key pairs can be performed prior to the auction



(a) 128 Bits



(b) 512 Bits

Figure 3.12: Verifications Sakurai & Miyazaki

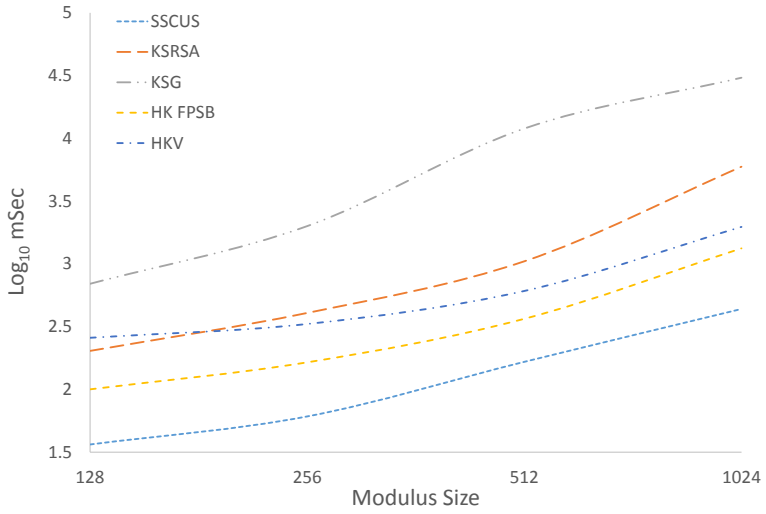


Figure 3.13: Initialization Times

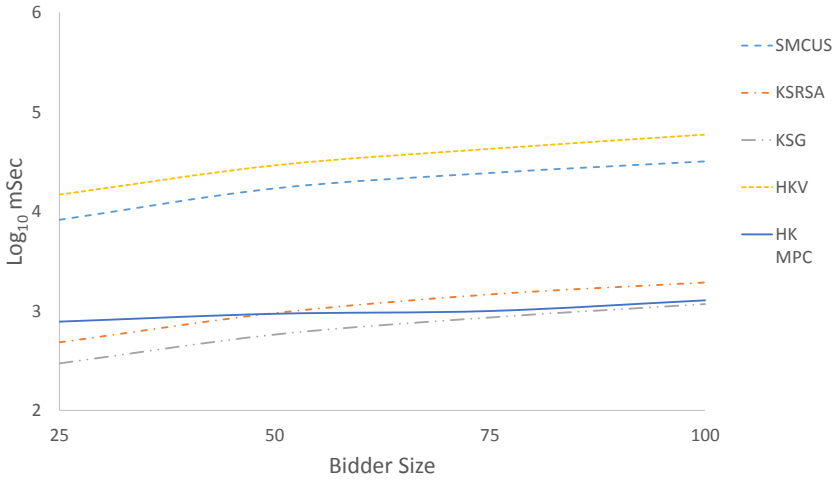


Figure 3.14: Verifications Times

starting. Also if the keys are created in parallel manner it would be significantly faster. Finally, the fastest protocol is Sakurai & Miyazaki protocol.

In the case of verification, the most computationally expensive protocol is HKV, and the fastest is Sako ElGamal. Hiroaki Kikuchi verifiable protocol is the most computationally expensive since we are try to reconstruct a polynomial using Lagrange for polynomials with up to 50 terms, and each coefficient has around 500 bits. Following this protocol is Sakurai and Miyazaki since for each round, determination of equality or inequality requires many calculations, and there could be many rounds. The fastest verification protocol is ElGamal decryption which comes close to HK naive implementation.

Table one neatly displays for comparison and summary the result of a 512 bit modulus, 100 bidders, and price range set to 50%. We can observe the same results as the graphs.

Table 3.1: Summary Table, Time (s)

512 Bits				
Protocol	***Bidders***	***Prices***	***Initialization***	***Verification***
HK	100	50%	0.3652	1.2792
HK Verifiable	100	50%	0.6089	59.3113
Sako ElGamal	100	50%	11.9817	1.1700
Sako RSA	100	50%	1.0491	1.9344
Sakurai & Miyazaki	100	50%	0.1657	31.8553

3.3 SOFTWARE IMPLEMENTATION CHALLENGES

Design of privacy preserving mechanism entails combining different cryptographic primitives, and advance algorithms. As a result, constructing a privacy preserving

protocol for financial markets is an inherently challenging task. Below are a number of challenges existing in many of the protocols in the literature, that were also encountered in the selected protocols when theory has to face practicality.

3.3.1 Big Integer Math and Prime Numbers

Most of cryptography, in order to be secure, is based around modular computation of numbers with well over 300 digits. One of the most costly computation, and likely to create memory overflow is exponentiation, which is closely related to multiplication. Compounding the effect of large integer math, is the fact that many protocols have a crucial first step of creating safe primes along with primitive roots of the field. For example, two commonly used asymmetric encryption, ElGamal, and RSA, are considered secure when using 1024 bits (more than 300 digits), or more recently 2048 bits (more than 600 digits). In many cases, a given protocol requires generating as many prime keys, and primitive roots, as the number of players participating. Since the implementation was in C++, and not in Java, or C# language, there is no out of the box big integer library. Designing a software package to manage computation of big integers, random number generation, prime number generation, and prime number factorization (used in algorithms that find generators of a cyclic group) is a project by itself. The solution provided above leverages the Integer Class provided by Crypto++, which is used by default in nearly all public/private key and general number theoretic operations. However, for the protocol by [27], specifically in the verifiable protocol, using the Integer Class would overflow during exponentiation since modular reduction was not an option during the exponentiation steps until the final reconstruction. For this case, the solution utilized the General Multiple Precision library offered for GCC compilers version 6.x.x and higher.

3.3.2 Concurrency in C++

All of the protocols that were selected required at some step(s) computation that can experience a benefit in performance if programmed with multiple threads. For instance, in the protocol by [60], the initialization step consist of generating an equal number of encryption keys as the number possible price choices. In the verification step, of the same protocol, one must decrypt as many bids as there players in each rounds. Both scenarios are ideal candidates for a multi-threaded or parallel computed model. The same can be concluded for the protocol [27] in the random polynomial generation step, a multi thread approach increases execution performance. The challenges is of course thread locks, are loss of sequential order, and thread guards.

CHAPTER 4

CONCLUDING REMARKS AND FUTURE WORK

In the previous sections we covered different security models, types of auctions, and security issues when designing an auction protocol. In addition, in Section 2, we reviewed novel ideas that have surged in the evolution of sealed-bid electronic auctions. As described before, typically, sealed-bid auctions consist of first price or second price. The method can be in an English style (increasing price) or Dutch style (decreasing price). Other auctions we covered here were combinatorial, and (M+1). However, a vast majority of the literature is dedicated to Vickrey and FPSB. Second price auctions are often subject to multiagent systems due to three important properties, namely, low bandwidth and time consumption, a dominant strategy to bid one's true valuation exists, and bids expressing private values remain secret. The literature also cautions the use of this economic mechanism. Undesired characteristics are described in [11] [10] [64] [9], but shortly, these include behavior of antisocial agents and antisocial strategies, lower revenue, bidder collusion, and lying in sequential auctions of interrelated auctions. One paper in particular [8], leads us to believe that there is no unconditional privacy in FPSB and Vickrey auctions if we consider protocols that do not rely on computational intractability assumptions or trusted third party (TTP). Currently a prevailing problem with all forms of auctions is the possibility of collusion. Under one scenario, bidders can coordinate to insert artificial prices such that a member pertaining to the group can obtain the reward at a favorable price. Another possibility is the collusion of bidder(s) and auctioneer(s) which can arise in many forms, one commonly in the form of bribery. Furthermore, auctioneers can collude to gain a higher revenue for themselves and for the sellers of the good. Even with

cryptographic protocols, most of the aforementioned papers suffer from a trust in auctioneer, and/or bidders. At the same time, many of the same protocols add additional time complexity and round complexity to ensure security of sealed-bid auctions, while some fail if players are tied, or if players deviate from protocol rules. For example, if we analyze the published work of [27],[3],[38] we see some of the problems mentioned. In [27] the auctioneers can frame the bidders as they are responsible for bidders ID and for the process of signing a received bid. At the time of bidding anyone can insert a fake bid, especially using a pseudonym, as there is no protocol for authentication. The protocol is easily disturbed by a malicious agent that insert a random bid value not present on the list of possible values that bidders are expected to choose because then the protocol can never assign a winner. In [3] values of the bids are not protected from the auctioneers, as there is to need to perform decryption over all bids to determine a winner. Auctioneers have the ability to accept a bid, and they are responsible for publishing it on the blackboard, however, there is nothing preventing them from colluding with another player and not publishing the bid. In addition, it is stated explicitly that the protocol has a problems when the agents supply no or false keys, called key denial problem. In [38] the procedure naturally reduces the total computational cost for bidders since they do not have to compute any proofs, instead the Auction manager does the heavy lifting. Using the procedure provides robustness against players trying to insert a malicious bid. One of the problems however, is that the secrecy of bids, as well as anonymity, is only guaranteed if both of the auction managers AM1 and AM2 do not collude, which is very optimistic, and unlikely, especially when there is a profit to be gained. The other problem is that there are two ambiguous cases which means that we cannot determine weather only a single person bid the highest value or if no one bid the highest value. In some auction protocols, for example, [27], [54], [29], [28] efficiency is improved by using homomorphic bid opening during bid reconstruction. An attack generated producing invalid bids can

have detrimental impact as shown in [51]. Specifically, an attack of invalid bids can affect quality of correctness and fairness. A possible solution is to impose a verification mechanism which then affects efficiency provided by homomorphic techniques.

In Chapter 3, an analysis of five different protocols in the literature for sealed-bid auctions were provided. The protocols consisted of different approaches. Namely, the focus was on protocols using multiparty computation, asymmetric key encryption, and commitment scheme. An emerging observation is that protocols using MPC will have a large number of communication rounds, and if the bid value is stored in the polynomial degree, it will result in increasing time complexity. In the case of using public key cryptosystem we can encrypt the bid and decrypt efficiently, however, we must realize that there is risk of auctioneers opening all bids since they hold all the decrypting keys. Commitment schemes also suffer from communication complexity since at every round bidders must prove that their commitment is not the standing price. In conclusion, a protocol can be fast like in the case of HK and Sako, but in order to provide higher security, and robustness such as HKV or Sakurai & Miyazaki, we must spend extra computation and/or communication rounds. To conclude, the expansion of computers and the internet means that e-commerce is an unavoidable piece of our society. In order, to face the issues of fair trading, and allowing auction economic mechanism to establish a fair price for goods, we must construct secure and robust anonymous protocols. While the literature is rich in novel ideas, there is still room for improvements in the area of computational efficiency [30], privacy in the active adversary model, and unconditional security. Hence the direction of new protocols is aiming to provide unconditional privacy, that is not relying on auctioneers, initializing parties, or third party members, while ensuring the qualities described in section 1.2, and improving round and time complexity. As part of future work, I would like to provide a novel approach to sealed-bid auction.

APPENDICES

APPENDIX A

PUBLICATIONS:

The results of this research appear in the following publications:

[1] Ramiro Alvarez, Mehrdad Nojoumian. *Comprehensive Survey on Privacy-Preserving Protocols for Securing the Financial Markets of Electronic Auctions.* Submitted and under review, 2018.

[2] Ramiro Alvarez, Mehrdad Nojoumian. *Efficient Implementation of Privacy-Preserving Protocols for Securing the Financial Markets.* Submitted and under review, 2018.

BIBLIOGRAPHY

[1] Masayuki Abe and Koutarou Suzuki. M+ 1-st price auction using homomorphic encryption. In *International Workshop on Public Key Cryptography*, pages 115–124. Springer, 2002.

[2] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 1–10. ACM, 1988.

[3] Joan Boyar, David Chaum, Ivan Damgård, and Torben Pedersen. Convertible undeniable signatures. In *Conference on the Theory and Application of Cryptography*, pages 189–205. Springer, 1990.

[4] Felix Brandt. Cryptographic protocols for secure second-price auctions. In *Workshop on Cooperative Information Agents*, pages 154–165. Springer, 2001.

[5] Felix Brandt. Secure and private auctions without auctioneers. *Tech Report*, 2002.

[6] Felix Brandt. A verifiable, bidder-resolved auction protocol. In *Proceedings of the 5th International Workshop on Deception, Fraud and Trust in Agent Societies (Special Track on Privacy and Protection with Multi-Agent Systems)*, pages 18–25. Citeseer, 2002.

[7] Felix Brandt. Fully private auctions in a constant number of rounds. In *International Conference on Financial Cryptography*, pages 223–238. Springer, 2003.

[8] Felix Brandt and Tuomas Sandholm. On the existence of unconditionally privacy-preserving auction protocols. *ACM Transactions on Information and System Security (TISSEC)*, 11(2):6, 2008.

[9] Felix Brandt, Tuomas Sandholm, and Yoav Shoham. Spiteful bidding in sealed-bid auctions. In *IJCAI*, volume 7, pages 1207–1214, 2007.

[10] Felix Brandt and Gerhard Wei. Vicious strategies for vickrey auctions. In *Proceedings of the fifth international conference on Autonomous agents*, pages 71–72. ACM, 2001.

[11] Felix Brandt and Gerhard Weiß. Antisocial agents and vickrey auctions. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 335–347. Springer, 2001.

[12] Christian Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 120–127. ACM, 1999.

[13] Chin-Chen Chang and Ya-Fen Chang. Efficient anonymous auction protocols with freewheeling bids. *Computers & Security*, 22(8):728–734, 2003.

[14] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[15] Ronald Cramer, Ivan Damgård, and Jesper B Nielsen. Multiparty computation from threshold homomorphic encryption. In *Int. Conference on the Theory and Applications of Cryptographic Techniques*, pages 280–300. Springer, 2001.

[16] Sven De Vries and Rakesh V Vohra. Combinatorial auctions: A survey. *INFORMS Journal on computing*, 15(3):284–309, 2003.

[17] Edith Elkind and Helger Lipmaa. Interleaving cryptography and mechanism design. In *International Conference on Financial Cryptography*, pages 117–131. Springer, 2004.

[18] Matthew K Franklin and Michael K Reiter. Verifiable signature sharing. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 50–63. Springer, 1995.

[19] 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.

[20] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *IJCAI*, volume 99, pages 548–553. DTIC Document, 1999.

[21] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[22] Louis C Guillou and Jean-Jacques Quisquater. A paradoxical identity-based signature scheme resulting from zero-knowledge. In *Proceedings on Advances in cryptology*, pages 216–231. Springer-Verlag New York, Inc., 1990.

[23] Michael Harkavy, J Doug Tygar, and Hiroaki Kikuchi. Electronic auctions with private bids. In *USENIX Workshop on Electronic Commerce*, 1998.

[24] Min-Shiang Hwang, Eric Jui-Lin Lu, and Iuon-Chang Lin. Adding timestamps to the secure electronic auction protocol. *Data & Knowledge Engineering*, 40(2):155–162, 2002.

[25] Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In *Int. Conference on the Theory and Application of Cryptology and Information Security*, pages 162–177. Springer, 2000.

[26] Ari Juels and Michael Szydlo. A two-server, sealed-bid auction protocol. In *International Conference on Financial Cryptography*, pages 72–86. Springer, 2002.

[27] Hiroaki Kikuchi. (m+ 1) st-price auction protocol. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 85(3):676–683, 2002.

[28] Hiroaki Kikuchi, Michael Hakavy, and Doug Tygar. Multi-round anonymous auction protocols. *IEICE Transactions on Information and Systems*, 82(4):769–777, 1999.

[29] Hiroaki Kikuchi, Shinji Hotta, Kensuke Abe, and Shohachiro Nakanishi. Distributed auction servers resolving winner and winning bid without revealing privacy of bids. In *Parallel and Distributed Systems: Workshops, Seventh International Conference on, 2000*, pages 307–312. IEEE, 2000.

[30] Sriram Krishnamachari, Mehrdad Nojoumian, and Kemal Akkaya. Implementation and analysis of dutch-style sealed-bid auctions: Computational vs unconditional security. In *1st International Conference on Information Systems Security and Privacy, ICISSP'15*, pages 106–113, 2015.

[31] Horng-Twu Liaw, Wen-Shenq Juang, and Chi-Kai Lin. An electronic online bidding auction protocol with both security and efficiency. *Applied mathematics and computation*, 174(2):1487–1497, 2006.

[32] Helger Lipmaa, N Asokan, and Valtteri Niemi. Secure vickrey auctions without threshold trust. In *International Conference on Financial Cryptography*, pages 87–101. Springer, 2002.

[33] John McMillan. Selling spectrum rights. *The Journal of Economic Perspectives*, 8(3):145–162, 1994.

[34] Toru Nakanishi, Daisuke Yamamoto, and Yuji Sugiyama. Sealed-bid auctions with efficient bids. In *International Conference on Information Security and Cryptology*, pages 230–244. Springer, 2003.

[35] Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 129–139. ACM, 1999.

[36] Khanh Quoc Nguyen and Jacques Traoré. An online public auction protocol protecting bidder privacy. In *Australasian Conference on Information Security and Privacy*, pages 427–442. Springer, 2000.

[37] Mehrdad Nojoumian. *Novel Secret Sharing and Commitment Schemes for Cryptographic Applications*. PhD thesis, Department of Computer Science, University of Waterloo, Canada, 2012.

[38] Mehrdad Nojoumian and Douglas R Stinson. Unconditionally secure first-price auction protocols using a multicomponent commitment scheme. In *International Conference on Information and Communications Security*, pages 266–280. Springer, 2010.

[39] Mehrdad Nojoumian and Douglas R. Stinson. Efficient sealed-bid auction protocols using verifiable secret sharing. In *10th International Conference on Information Security Practice and Experience, ISPEC'14*, volume 8434 of *LNCS*, pages 302–317. Springer, 2014.

[40] Hannu Nurmi and Arto Salomaa. Cryptographic protocols for vickrey auctions. *Group Decision and Negotiation*, 2(4):363–373, 1993.

[41] Baudron Olivier and Stern Jacques. Non-interactive private auctions. *Lecture Notes in Computer Science*, 2339:0354–0354, 2002.

[42] Kazumasa Omote and Atsuko Miyaji. An anonymous auction protocol with a single non-trusted center using binary trees. In *International Workshop on Information Security*, pages 108–120. Springer, 2000.

[43] Kazumasa Omote and Atsuko Miyaji. A second-price sealed-bid auction with verifiable discriminant of p 0-th root. In *International Conference on Financial Cryptography*, pages 57–71. Springer, 2002.

[44] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.

[45] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 248–259. Springer, 1993.

[46] David C Parkes, Michael O Rabin, and Christopher Thorpe. Cryptographic combinatorial clock-proxy auctions. In *International Conference on Financial Cryptography and Data Security*, pages 305–324. Springer, 2009.

[47] Torben Pryds Pedersen. Distributed provers with applications to undeniable signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 221–242. Springer, 1991.

[48] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing, 1998.

[49] Kun Peng, Colin Boyd, and Ed Dawson. A multiplicative homomorphic sealed-bid auction based on goldwasser-micali encryption. In *International Conference on Information Security*, pages 374–388. Springer, 2005.

[50] Kun Peng, Colin Boyd, and Ed Dawson. Optimization of electronic first-bid sealed-bid auction based on homomorphic secret sharing. In *International Conference on Cryptology in Malaysia*, pages 84–98. Springer, 2005.

[51] Kun Peng, Colin Boyd, and Ed Dawson. Batch verification of validity of bids in homomorphic e-auction. *Computer Communications*, 29(15):2798–2805, 2006.

[52] Kun Peng, Colin Boyd, Ed Dawson, and Kapalee Viswanathan. Efficient implementation of relative bid privacy in sealed-bid auction. In *International Workshop on Information Security Applications*, pages 244–256. Springer, 2003.

[53] Kun Peng, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Non-interactive auction scheme with strong privacy. In *International Conference on Information Security and Cryptology*, pages 407–420. Springer, 2002.

[54] Kun Peng, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Robust, privacy protecting and publicly verifiable sealed-bid auction. In *International Conference on Information and Communications Security*, pages 147–159. Springer, 2002.

[55] Kun Peng, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Five sealed-bid auction models. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, pages 77–86. Australian Computer Society, Inc., 2003.

[56] Kun Peng and Ed Dawson. Efficient bid validity check in elgamal-based sealed-bid e-auction. In *International Conference on Information Security Practice and Experience*, pages 209–224. Springer, 2007.

[57] Michael O Rabin and Christopher Thorpe. Time-lapse cryptography. 2006.

[58] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM, 1989.

[59] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinational auctions. *Management science*, 44(8):1131–1147, 1998.

[60] Kazue Sako. An auction protocol which hides bids of losers. In *International Workshop on Public Key Cryptography*, pages 422–432. Springer, 2000.

[61] 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 *Proc. International Workshop on Cryptographic Techniques and E-Commerce*, pages 180–187, 1999.

[62] Yuko Sakurai, Makoto Yokoo, and Koji Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 30–37. ACM, 2000.

[63] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial intelligence*, 135(1-2):1–54, 2002.

[64] Tuomas W Sandholm. Limitations of the vickrey auction in computational multiagent systems. In *Proceedings of the Second International Conference on Multiagent Systems (ICMAS-96)*, pages 299–306, 1996.

[65] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[66] Douglas R Stinson and Ruizhong Wei. Unconditionally secure proactive secret sharing scheme with combinatorial structures. In *International Workshop on Selected Areas in Cryptography*, pages 200–214. Springer, 1999.

[67] Srividhya Subramanian. Design and verification of a secure electronic auction protocol. In *Reliable Distributed Systems, 1998. Proceedings. Seventeenth IEEE Symposium on*, pages 204–210. IEEE, 1998.

[68] Koutarou Suzuki, Kunio Kobayashi, and Hikaru Morita. Efficient sealed-bid auction using hash chain. In *International Conference on Information Security and Cryptology*, pages 183–191. Springer, 2000.

[69] Koutarou Suzuki and Makoto Yokoo. Secure combinatorial auctions by dynamic programming with polynomial secret sharing. In *International Conference on Financial Cryptography*, pages 44–56. Springer, 2002.

[70] Koutarou Suzuki and Makoto Yokoo. Secure multi-attribute procurement auction. In *International Workshop on Information Security Applications*, pages 306–317. Springer, 2005.

[71] Moshe Tennenholtz. Some tractable combinatorial auctions. In *AAAI/IAAI*, pages 98–103, 2000.

[72] Chia-Chi Wu, Chin-Chen Chang, and Iuon-Chang Lin. New sealed-bid electronic auction with fairness, security and efficiency. *Journal of Computer Science and Technology*, 23(2):253–264, 2008.

[73] Makoto Yokoo and Koutarou Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 112–119. ACM, 2002.

[74] Shuo Zheng, Luke McAven, and Yi Mu. First price sealed bid auction without auctioneers. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 127–131. ACM, 2007.