8-2007

An Efficient Asynchronous Peer to Peer Auction using Yao Oblivious Transfer

Charles Lobo
Clemson University, chlobo@clemson.edu

AN EFFICIENT ASYNCHRONOUS PEER TO PEER AUCTION USING YAO OBLIVIOUS TRANSFER

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Charles Harry Lobo
August 2007

Accepted by:
Dr. Samuel Sander, Committee Chair
Dr. Richard Brooks
Dr. Ian Walker

ABSTRACT

Distributed electronic auctions are increasingly preferred over centralized electronic auctions today. The success of peer-to-peer file sharing networks has made distributed electronic auctions a possibility. Due to trust and conflict of interest issues with centralized auctioneer systems, multiple auctioneers in distributed roles are preferred. However, there is a possibility of auctioneer node collusion [16] and auctioneer-bidder collusion and auctioneer-seller collusion in such mechanisms.

To overcome these problems, a new peer-to-peer auction protocol [17] with auctioneers forming auctioneer groups has been proposed. This protocol keeps the auctioneers honest by ensuring that no single auctioneer in the group has absolute control over the auction process. But, it leads to multiple bid comparisons and thus increases redundancy. It also fails to enforce a secure bid comparison method and hence fails to provide privacy of bids. This thesis presents a modified version of this protocol where the oblivious transfer method [14] is used to solve the Yao millionaires' problem [22] that arises between two auctioneer groups when they have to compare bids. Additionally, a 2nd price mechanism in which only the second highest bid is known to all the auctioneer groups except for the auctioneer group which holds the highest bid, ensures that no unnecessary bid comparisons are made between auctioneer groups. Hence, the result is an efficient auction protocol which is iterative, asynchronous, 2nd price and based on a peer to peer mechanism.

DEDICATION

This work is dedicated to my parents and my sisters for all the support they have given me during the course of my engineering studies.

ACKNOWLEDGMENTS

First of all I would sincerely like to thank Dr.Sander, my advisor for giving me the opportunity to do research in the field of network security. His constant encouragement, timely advice and unconditional support have helped me develop my technical abilities and oratory skills.

I would like to thank Mr. Rolli whose research has got me interested in the field of secure auctions. Thanks for all the fruitful correspondence we had that has helped me write this thesis.

I would like to thank Dr. Brooks and Dr. Walker for their valuable inputs and for serving on my thesis committee.

I would also like to thank my friends and my family for the support and encouragement they have given me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

Auctions are one of the leading mechanisms today used for buying and selling goods which do not have a fixed price or value. Auctions have progressed from the ancient English, 1st price sealed bid, Vickery and Dutch auction mechanisms, to the modern electronic or internet version employed by Ebay, Amazon, etc.

Even though these internet auctions are quite popular, the protocols struggle to import all the advantages of the traditional auctions while providing modern advantages like asynchronous bidding, iterative bids and proxy bidding. Moreover, the disadvantages of the traditional auction mechanisms like bidder collusion and malicious auctioneers provide a constant challenge to electronic auction protocols. Hence, we need to realize an auction protocol which will integrate all the advantages of modern and traditional auction mechanisms while also being secure and efficient.

1.1 Motivation

The auction setup which Ebay employs basically includes a central server or a group of servers that handles all the incoming bids from the proxy agents and does the necessary processing of the bids that are received. Hence, there is only one central auctioneer who conducts the entire auction. This system is dependent on the functional and ethical reliability of the central server. A denial of service attack on the central server or a mere crashing of the server could potentially affect the outcome of all current auctions. Additionally, a malicious party, who may or may not be the bidder or seller, could get control of the central server and affect the outcome of all current auctions. Finally, the system requires all the bidders to trust the administrators of the central server, but when the administrators are paid

based on the final selling price, a conflict-of-interest is present, and user's trust can be easily betrayed if the auctioneer creates fictitious bidders or colludes with one or more sellers or bidders. Although the trusted, centralized auctioneer provides a simple system that decreases overhead, the number of disadvantages far surpasses the advantages.

This leads us to the logical conclusion that trust in the central server must be reduced. This can be achieved by splitting the central server into many independent peers. Hence, a single auctioneer is now replaced by multiple auctioneers who will combine together to perform the duties of the single auctioneer. This method helps in removing the disadvantages posed by the centralized auction method. The latest protocol for distributed auctions proposed by Rolli et al [17] provides a distributed mechanism for ascending second price auctions, but it requires multiple bid iterations, which increases overhead. Also, the bid comparisons are not done in a secure manner, and hence, the intermediate highest bids are not anonymous. Thus, the goal of this thesis is to effectively implement the distributed approach more efficiently while maintaining the anonymous nature of intermediate highest bids.

1.2 Thesis Overview

This thesis recognizes the bid comparison to be done between auctioneer groups as a Yao Millionaires problem [22] and solves it using oblivious transfer [14] to ensure secure bid comparison. It also implements second price intermediate higher bid comparison to decrease the bid iteration and improve efficiency.

The result is that an arbitrary precision can be achieved in a secure protocol which does not reveal the highest price at any time during the auction.

1.3 Structure of the Thesis

The thesis is structured as follows:

Chapter 1 gives the motivation and the thesis overview.

Chapter 2 gives an introduction to auctions. It gives a brief history of auctions and defines the different types of auctions that have been used through history. It also defines the various parameters which have to be fulfilled to design the ideal auction protocol.

Chapter 3 talks about the related work done in the field of distributed secure auctions. It focuses on the different auction protocols, their advantages and disadvantages based on the parameters set in chapter 2.

Chapter 4 discusses the modified auction protocol, the theory behind its proposal and the security analysis of the proposed protocol.

Chapter 5 talks about the implementation of the auction protocol. Its compares the results obtained with the results from an auction that does not use Yao oblivious transfer for bid comparisons. It also highlights the important conclusions that can be made after analyzing the auction protocol and discusses the future work that can be done to improve the performance of the proposed auction protocol.

CHAPTER TWO
INTRODUCTION TO AUCTIONS

Auctions are one of the leading mechanisms in the world of electronic commerce today. Auctions are used for buying and selling goods varying from cattle and food grains, to electronics. So popular is the method that it is even used by the government to sell radio frequency spectrum licenses. In today's world with the emergence of the internet, "electronic" or "internet" auctions have become an effective means to buy and sell new and used goods. As of February 2007 the total dollar value of merchandise sold in auctions is about $14.28 billion.

2.1 What is an Auction?

The word auction is derived from the Latin word "augere" which means "to increase". Thus, an auction is the process of buying and selling things by

a. Offering them up for Bidding.

b. Taking Bids.

c. Selling the item to the highest Bidder.

The common terminologies used for describing an auction and during the process of an auction are as follows:

a. Auctioneer: The auctioneer is the authority that performs the auction. Auctioneers are usually trained for their role and know the legal and practical aspects of conducting the auction. Sometimes it is required for the auctioneer to be licensed and bonded to conduct an auction.

b. Seller: The seller is the person who wants to sell an item owned by him using the auction mechanism. Generally, the seller is a person who is registered with the auction house and is known to all the potential buyers who will bid for the item being sold.

c. Bidder: A bidder is the person who offers an amount of money for the item being sold at a particular auction session.

2.2 History of Auctions

The earliest auctions have been known to have been conducted in Babylon in the year 500 B.C in which marriageable women would be sold on the condition of marriage. Beautiful women would fetch higher bids while less beautiful women would have to pay a dowry to be accepted into the auction. Thus, the bidding price would be negative which means that the bidders would receive money for marrying such women. Auctions can also be dated back to the Romans in the year 193 A.D in which literally the entire Roman Empire was on auction.

A few hundred years down the line in Great Britain the most prominent and classic auction houses emerged in the form of Sotheby's, Christie's and Bonham's in 1744, 1766 and 1793, respectively. In America, auctions were used to sell second hand goods, domestic animals and slaves. Also, in the nineteenth century auctions were used to sell vegetables, fruits and flowers in Holland and fish in Germany.

Today with the emergence of the internet there has been a rising amount of users who use electronic auctions (see section 2.4). Auction sites such as eBay, Amazon and GoIndustry.com are popular alternatives for people who want to trade goods over the internet.

2.3 Traditional Auction Types

The auctions mechanisms have evolved over time to incorporate maximum security while also adjusting to the markets conditions of particular countries. As seen in the sections below the auctions mechanisms have made some typical contributions to the modern world of electronic auctions. The difference between the traditional auction types have been summarized in the Table 2.1.

2.3.1 English Auction

In an English auction the participants bid openly against each other with each bid being higher than the previous bid. Initially the seller sets a minimum price for the item being sold. If any bidder fails to bid an amount which is equal or higher than the minimum price set then the item remains unsold. The auction ends when no participant is willing to place a higher bid or the predetermined buy-out price is reached. This type of auction is used in English auction houses like Sotheby's, Christie's and Phillips. It is also known as the "ascending open-cry" auction.

Table 2.1 Traditional Auction Types

Auction Type =>	**English**	**First-Price Sealed Bid**	**Vickery**	**Dutch**
Bidding Type	Iterative, Public	Single, Secret	Single, Secret	Iterative, Public
Winning Bid	Highest Bid	Highest Bid	Second Highest Bid	When Bidding stops

2.3.2 1st-Price Sealed Bid Auction

In a 1st price sealed bid auction, all the bidders simultaneously submit a single sealed bid for the item. The highest bidder wins the auction and pays the amount he bid. It is also known as "Sealed High-Bid" auction.

2.3.3 Vickery Auction

The Vickery auction was proposed by Nobel Prize Laureate William S. Vickery in 1961 [21]. It is also known as "Sealed Bid 2nd price" auction. It works in a similar manner to 1st price sealed bid auction except that the winning bidder pays the 2nd highest bid rather than his own.

2.3.4 Dutch Auction

In the Dutch auction the auctioneer starts with a high price and decreases the price until a bidder is willing to pay the amount or until the predetermined minimum is reached upon which the good is not sold. The name comes from the selling of Dutch flowers where an electronic buzzer connected to a clock is used to implement the auction. These auctions are suitable for selling perishable items which lose their value during the course of the auction. Some forms of the Dutch auction were used to sell clothes in USA.

2.4 Electronic Auctions

Ebay [6] is one of the most popular electronic auction sites on the internet today. As of February 2007, the number of auction listings shows a figure of 588 million. Also, the total dollar value of merchandise sold in the auctions is about $14.28 billion. The majority of the internet auction sites have auction mechanisms that are roughly based on the traditional English auction mechanism. But due to its electronic nature the bidding is done via "proxy agents" which submit increasing bids on behalf of the actual bidder. Hence, the bidder does

not have to be online all the time especially since a particular auction session can go on for many days.

2.5 Requirements for the ideal auction protocol

An auction protocol can be considered as an ideal auction protocol if it fulfills certain requirements. These requirements [17] have been identified by Rolli et al based on the role that the auctioneer plays and the bidding process itself.

2.5.1 Auctioneer Requirements

The auctioneer needs to fulfill the following requirements

a. Second-price

Second-price auctions are generally preferred over first-price auctions. This is because in first-price auctions multiple bidders can collude in such a way that all the bidders bid substantially smaller amounts such that the highest bidder can win the auction at a much smaller price than he would have to pay. After winning the auction the highest bidder can share his savings with the colluding bidders. This situation is eliminated in second-price auctions. This is because the highest bidder has to bid his true evaluation of the product and the colluding bidders cannot decrease their evaluation and obtain the good without making a loss. Hence, second-price auctions have been said to be "self-enforcing" [16]. Hence, the auctioneers participating in the auction will jointly determine the final price which will be the second highest bid amount plus an increment [17].

b. Secret highest bid and identity of bidder

The identity of the highest bidder and his bid should remain secret at any moment in the auction. If the current bid is above the minimum bid for the particular good then it could potentially be the highest bid. Hence, even if a particular bidder highest bid is revealed to an

auctioneer no other bidder should know if it's the highest bid. This information should remain only with the participating auctioneers.

c. Resistant to bidder exclusion

Bidders that have registered for the auction should be able to participate in the auction at all times. No auctioneer or group of auctioneers should have the power to illegally drop any bidder from the auction or drop and bid from the bidder.

d. Robust to paralysis attacks

The auction could involve one or multiple auctioneers. Hence it is possible that a few auctioneers could go offline due to problems like denial of service attacks, servers crashing, natural calamity or too much traffic and paralyze the entire auction. But, the auction protocol should be able to withstand such loss of auctioneers and be able to function properly despite this loss. Also, the protocol should be able to withstand any kind of deliberate blocking done by a single auctioneer or a group of auctioneers.

2.5.2 Bidding Process Requirements

The Bidding process needs to fulfill the following requirements

a. Spontaneous bidder entry and exit

The auction protocol should be able to add bidders to the auction as and when the bidders want to join. Hence, the protocol should give unrestricted access to the new bidders joining the auction and accept bids being put up by them without affecting the auction process itself. Also, the bidders should be allowed to exit the auction as and when they feel like.

b. Iterative

The final result of the auction should be decided over several iterations where bidders put several bids, before the time limit or maximum amount limit and get sufficient feedback of the progress of the auction.

c. Asynchronous Bids

Bidders should be permitted to submit bids at any time during the auction. This helps the bidder as he does not have to be online during the entire auction. The bidder should be able to put in his group of bids and stay offline till the result is decided. Hence, the protocol should be able to achieve a system similar to "proxy bidding".

d. Single Independent key-pair

Each bidder should need only one pair of keys for the entire auction. Thus, each bidder should have a public and a private key that is generated before or during the auction process. This key-pair should be reusable and should be based on modern cryptographic techniques.

e. Non-repudiation

Every bid should be accountable. It should not happen that a bidder wins an auction and disputes his own bid. This can be achieved by authenticating every bid submitted by each bidder.

CHAPTER THREE
RELATED WORK

3.1 Early Distributed Auctions

The Vickery [21] auction or second-price auction format is considered as the ideal auction format. While this works well for sealed bid auctions it has some disadvantages when implemented in iterative auctions. The disadvantages include the vulnerability against a malicious auctioneer [18] and the reluctance of bidders to truthfully reveal their private information [14]. These problems are addressed by Brandt [3] in his work on a secret sharing scheme which does not have any auctioneers, but it enables bidders to jointly compute the final price. It is one of the first methods that look at distributing the role of the single auctioneer, which in this case is done by the bidders themselves. But, since this mechanism looks into selling M identical items on a non-iterative basis the desired iterative requirement is not fulfilled. Franklin and Reiter [7] also proposed a sealed bid mechanism which was based on distributed auctioneers using verifiable signature-sharing. But, the protocol cannot be extended to iterative bidding. Also, at the end of the auction all the bids are known to all the auctioneers.

Brandt [4] later proposed an iterative adaptation to his previous approach. But since the protocol uses round-wise bidding it fails the requirement for the bids to be asynchronous. Also, since the bidders do all the computation they have to stay online for the duration of the auction. This could cause a major problem on the internet as the bandwidth of bidder peers, which is not necessarily large, could slow down the auction since the each bidder has to wait for the previous bidder to submit his bid.

Kikuchi [13] developed an electronic auction with distributed auctioneers, which is based on multiparty secret computation. But, it faces the problem that it requires a threshold number of obedient auctioneers to perform the auction. If the number of obedient auctioneers goes below this threshold value the auction can be compromised.

Other methods include those which have a partially trusted third party. Cachin [5] proposed a method in which the third party or auctioneer is totally oblivious to the bidding amounts and only does the computation on the bids using cryptographic means. He introduces the outstanding problem called "Millionaires Problem" [22] in which two millionaires want to know who is richer without revealing their riches to each other. It provides bid privacy as in the final bid and the identity of the winner is not revealed. But, the problem again arises in that it supports only sealed bid format and not the iterative format. Also, having a partially trusted third party is similar to the concept of having a centralized auctioneer which is like "Putting the fox in charge of the hen house" [19].

3.2 An asynchronous and secure ascending peer-to-peer auction [17]

Rolli et al [17] proposed a distributed mechanism for ascending second price auctions. The method includes an auction setup in which the seller initiates the auction and selects the auctioneers using addressing mechanisms in structured peer-to-peer networks like CAN [19] and Chord [20] thus ensuring randomness in the selection of auctioneers and subsequent auctioneer groups. A detailed explanation of the setup process is given in the Section 3.2.1.

3.2.1 Auction Entities

The auction consists of three different types of entities namely seller, bidder and auctioneer. Before the auction is setup these entities will be nothing but simple peers or nodes in the network. Hence, these peers will be spread throughout the network before the

auction. The different entities will perform different roles and therefore have different requirements. The seller only needs to initiate the auction and hence has to only communicate with the potential auctioneer nodes. Hence, the seller node does not need to have a very high capacity or constant connectivity. Similarly, the bidder nodes only need to register in the auction and prepare their bids and submit them. Hence, they do not need to exchange any kind of data or do any computation on data. Hence, the bidder nodes also need not have a very high capacity or constant connectivity. But, the auctioneer nodes need to communicate between each other and exchange data between them during the course of the auction. They need to be constantly online and hence need a constant connectivity with a high capacity to securely exchange data between each other. Hence, such nodes need to be similar to super-nodes [12]. Fig 3.1 shows the transformation of simple peers before the auction setup to the auction entities after the auction is setup.

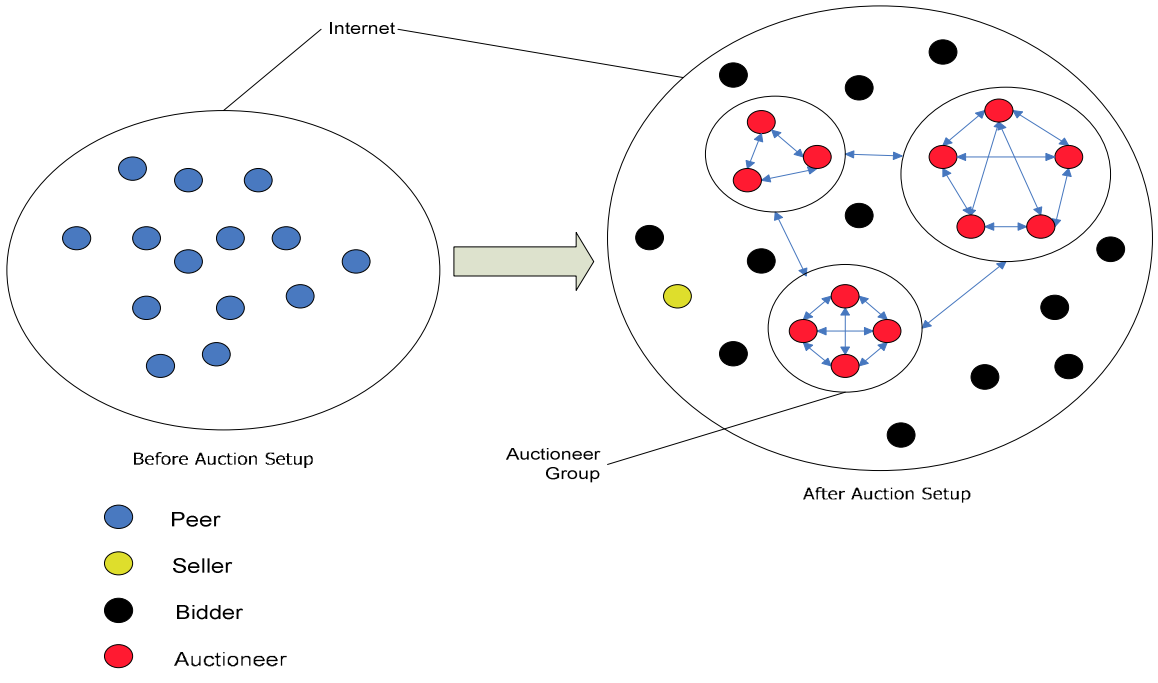

Figure 3.1 Auction entities [17]

3.2.2 Auction Initialization

As the auction does not intend to include any auction authority, the auction initialization is done by the seller himself. The seller creates the auction document D. This document has all information about the auction including the description of the items for sale, the particular auction type that will be used and the starting price of the auction.

The next important step in the auction initialization process is the distribution of the auctioneers into groups. This also has to be done by the seller. The following calculation is used to determine this distribution of auctioneers

$$G_{size} = [\log_X (A_{total})]$$

$$G_{number} = [\frac{Atotal}{Gsize}]$$

where, A_{total} – The total number of auctioneers participating in the auction.

G_{size} – The size of an auctioneer group.

G_{number} – The number of auctioneer groups in the auction.

The process of selection of the auctioneers is shown in the Fig 3.2. Initially the seller has knowledge of the available auctioneer nodes. The seller sends a participation broadcast to all the auctioneer nodes. Each auctioneer node acknowledges its participation in the auction by sending the hash value associated with the location or address of the node. Since, the selection of auctioneers into groups is done by the seller himself, it is possible that the seller can select collaborating or malicious nodes in the auctioneer groups. To avoid such a situation the hash value is used which is mapped to the location of the nodes using

mechanisms that are used in structured peer-to-peer networks like CAN [15] and Chord [20]. The address of a particular auctioneer can be found using the following formula:

$$\text{Address}(A_{ij}) = H_{i.Gsize+j}(D) \qquad \text{where } 0 \le i < G_{number} \text{ and } 0 \le j < G_{size} \qquad \ldots.(1)$$

A random function is used to distribute these hash values into groups, and hence, the seller effectively distributes the auctioneer nodes into auctioneer groups without any knowledge of the actual location or address of the nodes. A few bits in the hash value can also be dedicated to the location of auctioneer nodes based on continent or region such that the auctioneer groups selected have maximum physical separation. After the selection of the auctioneer groups the auction document D is modified with information of the auctioneer groups and sent over to all the participating auctioneer nodes.

After the formation of auctioneer groups, the auctioneers within a group communicate between each other and generate a common group public/private key pair. This key pair can be generated by one auctioneer within the group and sent to the other auctioneers, or a distributed key pair [1] can also be generated among the auctioneers.

Hence, at the end of the auction setup we have the following information:

G_i – Group i of auctioneers, where $0 \le i < G_{number}$,

K_i – Public key of auctioneer group G_i , and

A_{ij} – Auctioneer j of group i, where $0 \le i < G_{size}$.

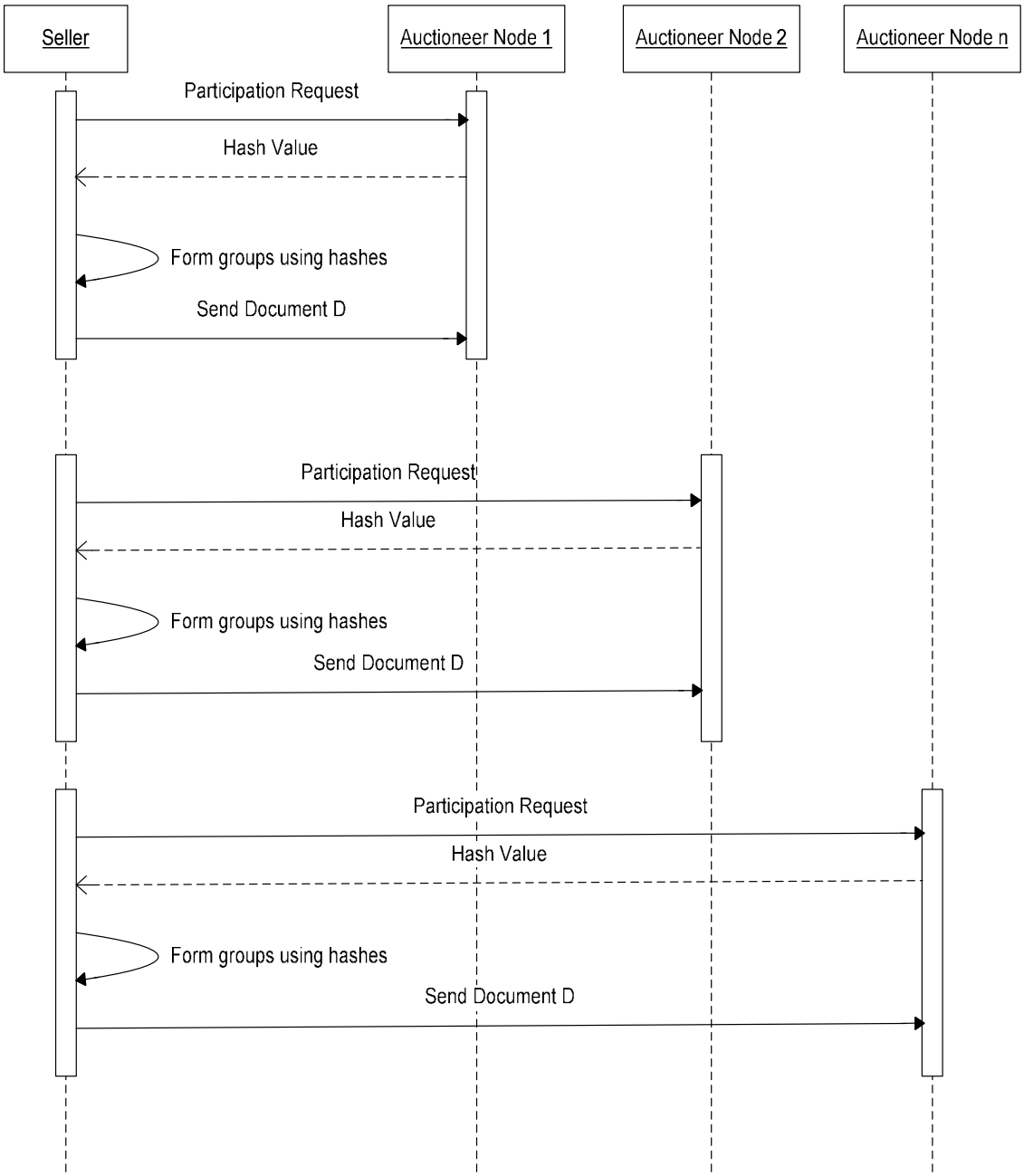

Figure 3.2 Auction setup process

3.2.3 Bidders joining the auction

In order to enter the auction a potential buyer has to procure the auction document D. The seller will setup a web server to advertise the auction. Potential buyers can visit the web server and obtain information about the product being auctioned. If the buyers are interested they will download the document D to get further information about the auction. The bidder can join the auction by placing a bid. This is done by the bidding process which is explained in Section 3.2.4.

3.2.4 Bidding process

Once the bidder obtains information of the product which is being sold he prepares a valuation v for the product. This value v is his true valuation of the product. The entire bidding process is shown in Fig 3.3. At any point in the auction the bidder selects a value p such that $p < v$ and $p >$ *current bid*. He then formulates a series of values which are monotonously increasing, up to the value of p. These values form what is known as a bid chain BC while the individual values are known as bid steps. To ensure non-repudiation the bidder digitally signs each bid.

The bidder can reproduce the address of any particular participating auctioneer using the formula (1) in Section 3.2.2. The bidder then chooses a hopping sequence of auctioneer groups H S as defined below:

$H\,S_i = j \quad$ where $0 \leq j < G_{number}$

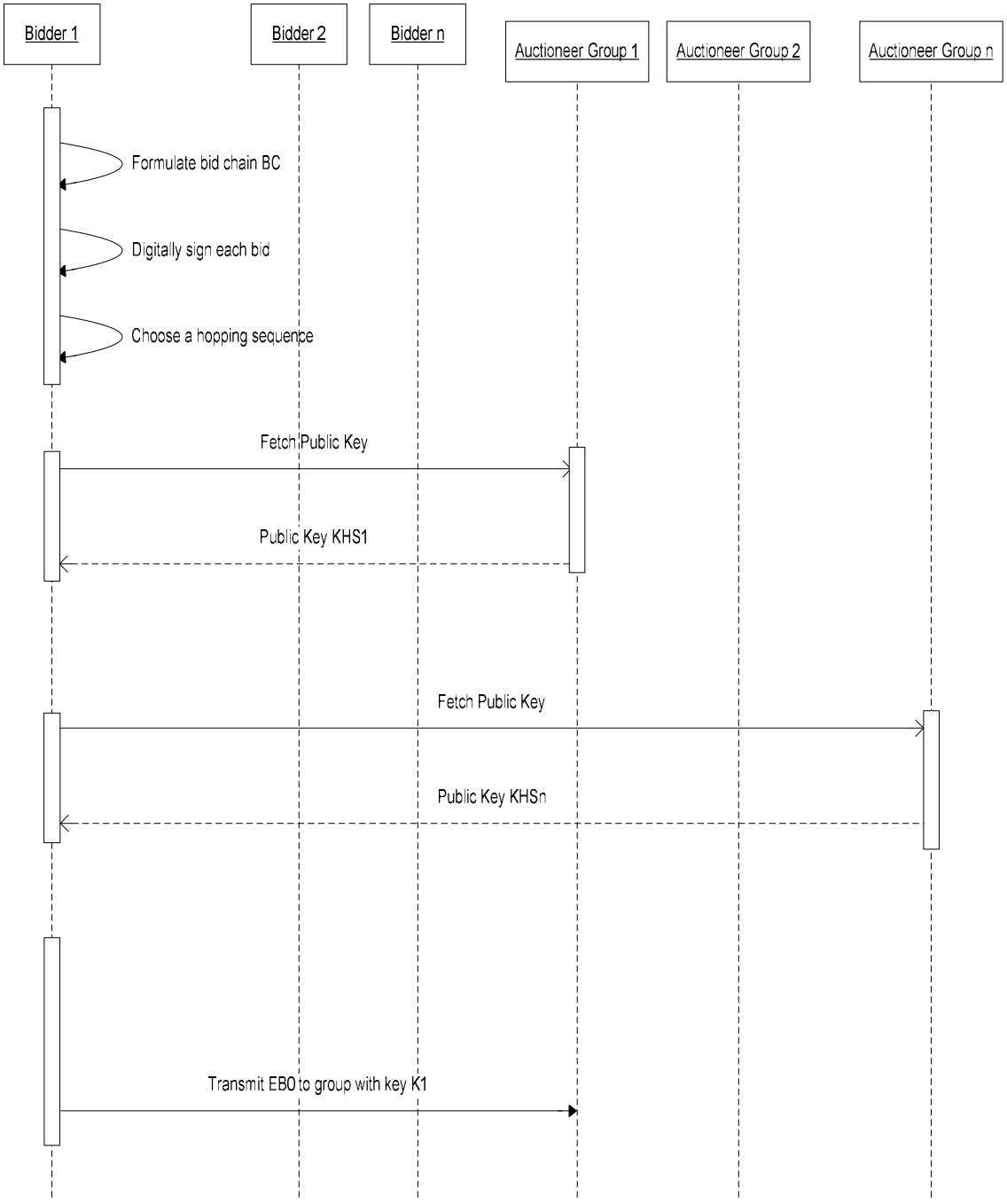

Figure 3.3 Bidding process

The bidder then fetches the public keys of the auctioneer groups he selects to send his bids to. The bidder then encrypts the highest bidding step with the first public key K_{HS0} of the hopping sequence. He then encrypts his second highest bidding step, the hash value of the highest bidding step and the encrypted highest bidding step with the second public key K_{HS1} of the hopping sequence. He repeats this procedure until all the bidding steps are encrypted and his lowest bid or starting bid is the last encrypted bid as shown below

$$E\ B_n = E_{HS0}(S_L(B_n))$$

$$E\ B_{n-1} = E_{HS1}\ (S_L(B_{n-1}), S_L(H(B_n)), E\ B_n)$$

. . . . . .

$$E\ B_1 = E_{HSn-1}\ (S_L(B_1), S_L(H(B_2)), E\ B_2)$$

$$E\ B_0 = S_L(H(B_1)), E\ B_1$$

where $E\ B_i$ = Encrypted bid B_i using public key K_i ,

S_L = Digital signature of bidder L , and

$H(B_i)$ = Hash value of the bid B_i .

Thus, the bidder L will start his bidding process by sending $E\ B_0$ to the auctioneer group which has the key K_{HSn-1}. Hence, for any auctioneer group to get the next bid step of bidder L the previous bid step has to be decrypted by the auctioneer group which receives the previous bid step as illustrated in Fig 3.4 and Fig 3.5.

Fig 3.4 shows an illustration of multiple layers of encryption used in this protocol. We see that the highest bid is encrypted first and later on the lower bids are encrypted which resembles the layers of an onion [11]. Hence, out of a total of 4 bids when the first layer of encryption is removed it reveals the lowest bid or 4th highest bid of the bidder and when the second layer of encryption is removed the 3rd highest bid is revealed. Hence, every layer of the onion is a layer of encryption which when removed gives the next highest bid step submitted by the bidder.

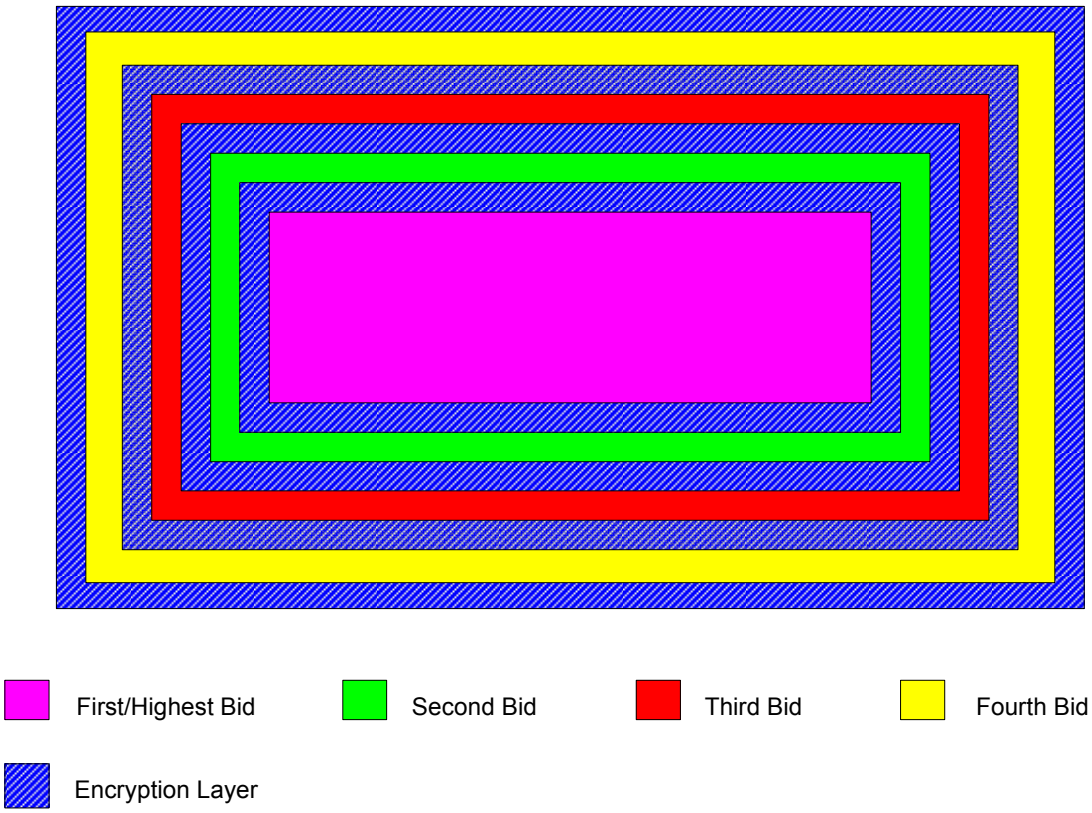

Figure 3.4 An illustration of multiple layer encryptions

A typical bid chain creation is illustrated in Example 3.1

Example 3.1 Consider the Fig 3.5 in which a bidder wants to submit a series of bids in ascending order. He wants to start off with an initial bid of $10 and monotonically increase to $12, $15 and finally ending at $30 which is his highest bid. Hence, he starts creating his bid chain by first encrypting his highest bid i.e. $30 by the public key of the 1st auctioneer group from the selected hopping sequence H S. In the next step he encrypts his second highest bid and the encrypted first highest bid with the public key of the 2nd auctioneer group from the hopping sequence H S and so on. Hence, we have a series of bids sandwiched between multiple layers of encryption as shown in Fig 3.5.

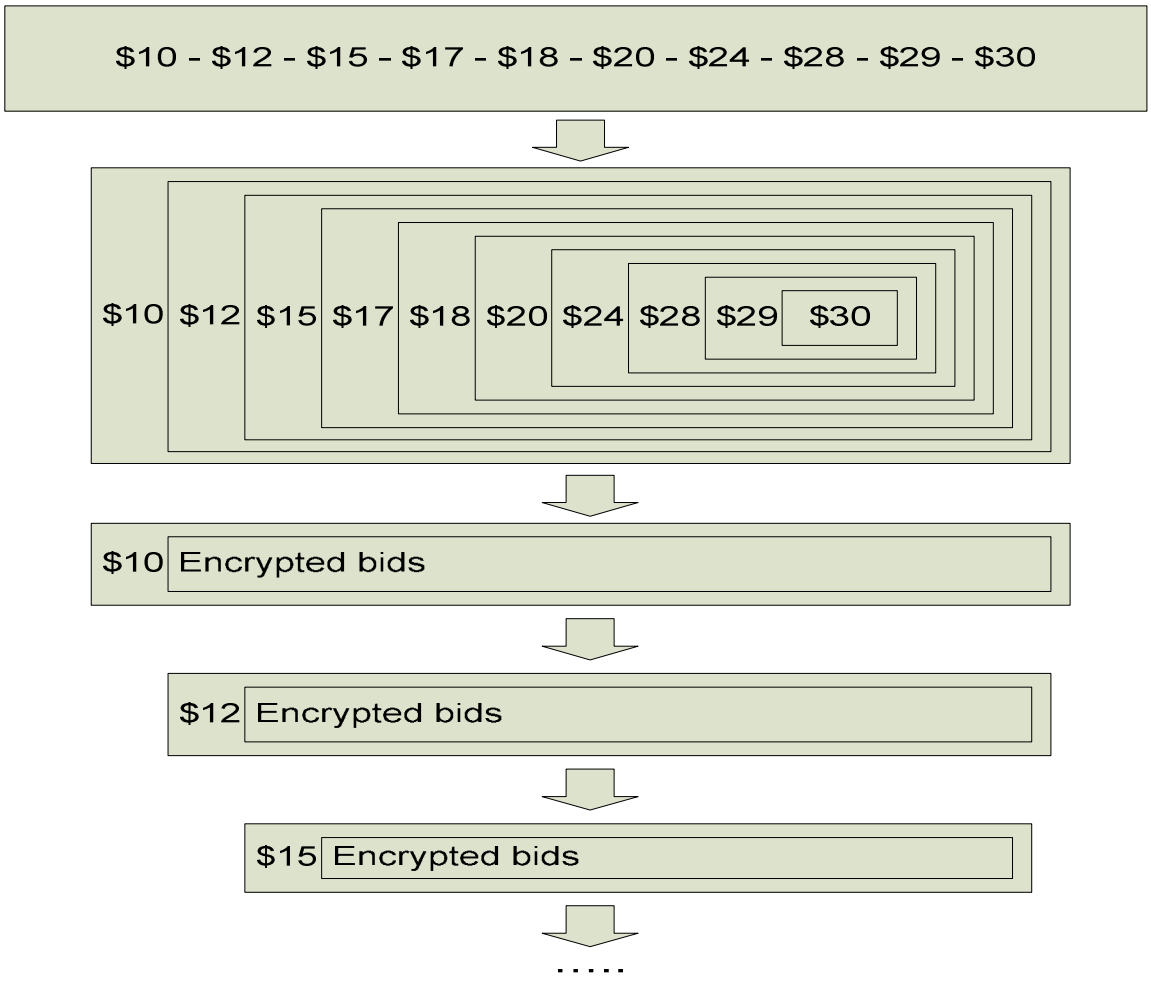

Figure 3.5 An illustration of formation of a bid chain [17]

3.2.5 Processing a Bid

When an auctioneer receives a bid chain E B_0 he first checks if their group key is same as the encryption key used for encrypting the bid E B_1. If this check is successful the bid E B_0 will be propagated to all the members within the group, else it will be forwarded to the applicable group. A randomly selected subset of auctioneers will then decrypt the bid E B_1 and the signed bidding step $S_L(B_1)$ will be propagated to all the members within the group to verify the validity of the bid by comparing the hashed value of the bid with the hashed value $H(B_1)$ which is included in the bid step. This is illustrated in Fig 3.6.

Once the bid is received the auctioneer group checks if the bid is greater than the standing highest bid HB_0 .i.e. $B_1 > HB_0$ **.** The value of HB_0 is known to all the auctioneer groups. But the bidder who has of the standing highest bid is not known to any auctioneer groups except for the auctioneer group that received the bid. If the bid B_1 is lesser than the standing highest bid HB_0 the group will transfer $S_L(B_1)$, $S_L(H(B_2))$ and E B_2 to the next auctioneer group in the hopping sequence. If the bid B_1 is greater than the standing highest bid HB_0 the auctioneer group will convey the auction status to the bidder and broadcast B1 as the standing highest bid.

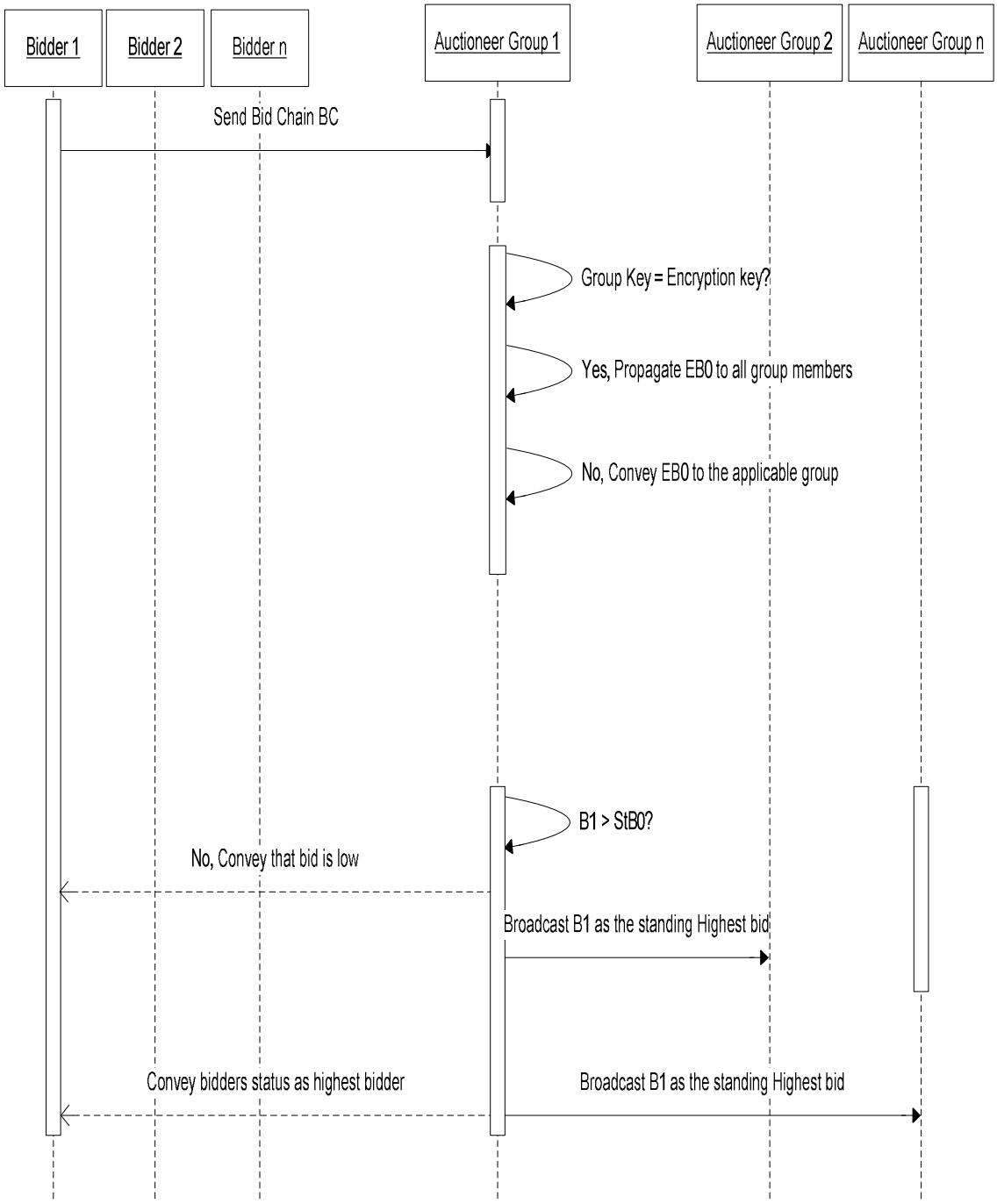

Figure 3.6 Processing a bid

This method fulfills most of the requirements for an ideal auction protocol. It supports an asynchronous, iterative proxy bidding format and ensures privacy of bids including the winning bids and the winning bidders' identity. But, the protocol does not talk about the possibility of deducing the bidders highest evaluation based on the size of the bid chain. Also, the system fails to implement the comparison of intermediate bids between auctioneer groups in a secure manner. Hence, it is possible that during comparison of the current bid and highest bid the auctioneer groups will know the values of the two bids that are being compared, and hence, the winning bid will be known to more than one auctioneer group. Thus, the protocol fails to ensure complete privacy of the highest bid. If the bid comparison is done using secure means it would also lead to many unwanted exchanges between the two auctioneer groups for each comparison thus making the protocol highly inefficient. Since, the auction protocol uses a first price mechanism during the intermediate bid comparisons, a comparison would have to be done between auctioneer groups for every bid that is received as shown in the Example 3.2.

Ex 3.2 Consider the case where there are m=100 auctioneer groups and n=10 bidders. Let each bidder put a maximum of k=50 bids in the entire auction. Also, consider that all the bids are exhausted to know the outcome. Hence we have the following:

Total Number of bids in the auction $b = n * k = 500$.

Now suppose the number of handshake exchanges used for the secure comparison is $h = 3$.

To do a secure comparison for each bid we will require $(b-1) * h = 499 * 3 = 1497$ exchanges between auctioneer groups. Hence, for a million bids the auction the time taken for secure exchange will be 3 million.

This would make the protocol highly inefficient. Hence, we need a mechanism which will decrease the number of rounds or exchanges done between auctioneer groups during the course of the auction.

The auction protocol proposed in this thesis is based on the method proposed by Rolli et al. But, it deviates at the point where the comparison between bids is done in a secure manner using cryptography. In particular this thesis successfully implements oblivious transfer method [14] between auctioneer groups and solves the "Millionaires problem" [22] that arises between the two auctioneer groups. Also, the current highest bid is never declared in the auction. For every step the second highest bid amount is made known to all the auctioneers. The advantage of this adaptation can be explained with the Example 3.2.

Example 3.2 Consider the figures from the Example 3.1. Hence, we have

Total Number of bids in the auction $b = n * k = 500$

But now instead of a secure comparison for every bid we only need to do the secure comparison if a bid is higher than the standing highest or 2^{nd} highest bid.

This modification reduces the number of secure comparisons done between the auctioneer groups and hence improves efficiency.

CHAPTER FOUR
AUCTION PROTOCOL WITH YAO OBLIVIOUS TRANSFER

4.1 Introduction

Systems like Kazaa [12] and GNUtella [8] are working proof of the success of decentralized peer to peer systems on the internet today. Now, the existing auction mechanisms depend on the centralized auctioneer or auction authority. But such systems require the bidders to trust the auctioneer or auction authority. To distribute this trust, various distributed auction protocols based on peer to peer mechanism have been proposed. This thesis tries to realize an auction protocol that is based on the peer-to-peer mechanism and can function without requiring any major memory requirements and provide a high efficiency coupled with adequate security.

4.2 Background

The protocol proposed in this thesis uses the 2nd price mechanism for comparison of bids. When an auctioneer group receives a particular bid it compares the bid to the 2nd highest bid of the auction. Since, the 2nd highest bid is the amount which is fairly close to the highest bid amount it is fair enough to compare a new bid with it. Hence, only those bids which are greater than the 2nd highest bid are subject to secure comparison with the auctioneer group that has the highest bid. Hence, the number of rounds of secure comparisons is decreased. The auction setup mechanism and bidding mechanism is the same as the Rolli et al [17] implementation that can be seen in Section 3.2. The processing of a bid varies from the Rolli et al [17] method and is explained in Section 4.2.1.

4.3 Protocol Enhancements

The proposed protocol basically adds the Yao oblivious transfer to compare bids between two auctioneer groups without them knowing each others bids. It also uses the to ensure privacy of intermediate highest bids.

4.3.1. Processing a bid

When an auctioneer receives a bid chain E B_0 he first checks if their group key is same as the encryption key used for encrypting the bid E B_1. If this check is successful the bid E B_0 will be propagated to all the members within the group else it will be forwarded to the applicable group. A randomly selected subset of auctioneers will then decrypt the bid E B_1 and the signed bidding step $S_L(B_1)$ will be propagated to all the members within the group to verify the validity of the bid by comparing the hashed value of the bid with the hashed value $H(B_1)$ which is included in the bid step. This is illustrated in Fig 4.1.

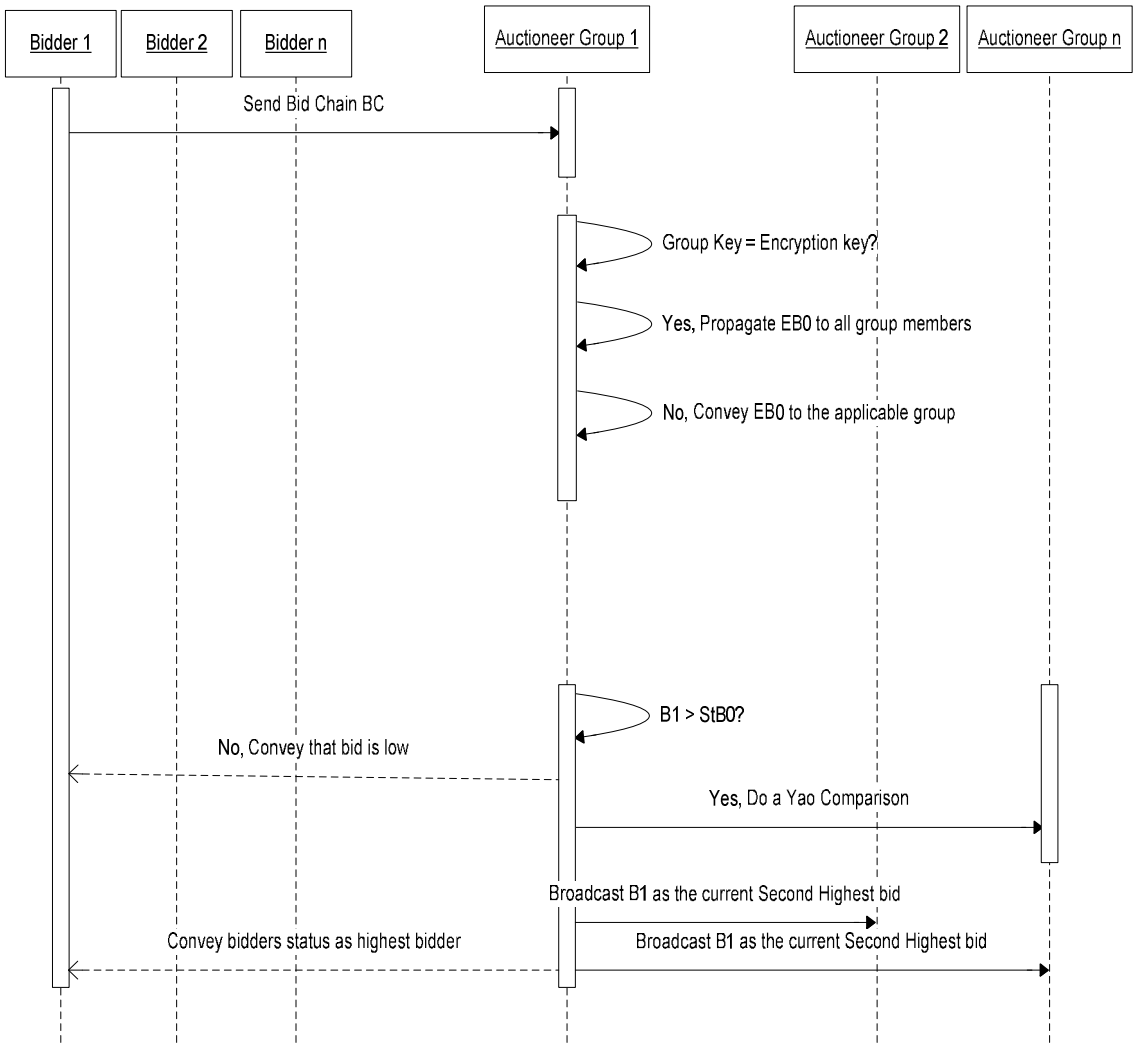

Figure 4.1 Processing a bid using Yao Oblivious Transfer

Once the bid is received the auctioneer group checks if the bid is greater than the standing highest bid StB_0 which is also the second highest bid of the auction .i.e. $B_1 > StB_0$. The value of StB_0 is known to all the auctioneer groups. But the bidder who has of the standing highest bid is not known to any auctioneer groups except for the auctioneer group

that received the bid. Also, the auctioneer group holding the highest bid is known to all the auctioneer groups participating in the auction. If the bid B_1 is lesser than the standing highest bid StB_0 the group will transfer $S_L(B_1)$, $S_L(H(B_2))$ and E B_2 to the next auctioneer group in the hopping sequence. If the bid B_1 is greater than the standing highest bid StB_0 the auctioneer group initiates a Yao transfer [22] with the auctioneer group that holds the highest bid.

The Yao transfer is used to determine if the current bid B_1 is greater than the current highest bid HB_0. If $B_1 > HB_0$, the auctioneer group will create a confirmation for the bid B_1 and send this confirmation to the bidder L. The auctioneer group will then broadcast the value HB_0 as the standing highest bid and update its status in the auction as the auctioneer group that holds the current highest bid. If $B_1 < HB_0$ the group will convey to the bidder that the bid is too low. The group will also transfer $S_L(B_1)$, $S_L(H(B_2))$ and E B_2 to the next auctioneer group in the hopping sequence. The Yao transfer is explained in Section 4.2.1.1.

4.3.1.1 Yao's Millionaires' problem [22]

Consider two millionaires Alice and Bob. Alice has i millions and Bob has j millions. Both want to find out which amount is greater without actually revealing their riches to each other. This classic problem is called the Yao's Millionaires' problem.

The same situation occurs between the auctioneer group that has the highest bid HB_0 and the auctioneer group that has a bid B_0 that is greater than the standing highest bid. The solution is explained in Section 4.3.1.2.

4.3.1.2 Oblivious transfer [14]

The solution to the problem mentioned above can be explained in terms of oblivious transfer also known as secure computation. Oblivious transfer is also regarded as the "crypto gate" for secure computation. It can be illustrated from Fig 4.2 .

The protocol is as follows:

1. To initiate the transfer each auctioneer group needs to have a public key and a private key. Group A has the bid amount B_1 while group B has the current highest bid HB_0.
2. Group A chooses a large random number, x and encrypts it with Group B's public key.

   $c = E_b(x)$

   where, E_b = Group B's public key
3. Group A computes $(c- B_1)$ and sends the result to Group B.
4. Group B computes the following 100 numbers

   $Y_u = D_b(c- B_1-u)$, for $1 \leq u \leq 100$

   D_b = Decryption algorithm with Group B's private key.

5. Group B chooses a large random prime p and computes the following 100 numbers:

   $Z_u = (Y_u \bmod p)$, for $1 \le u \le 100$

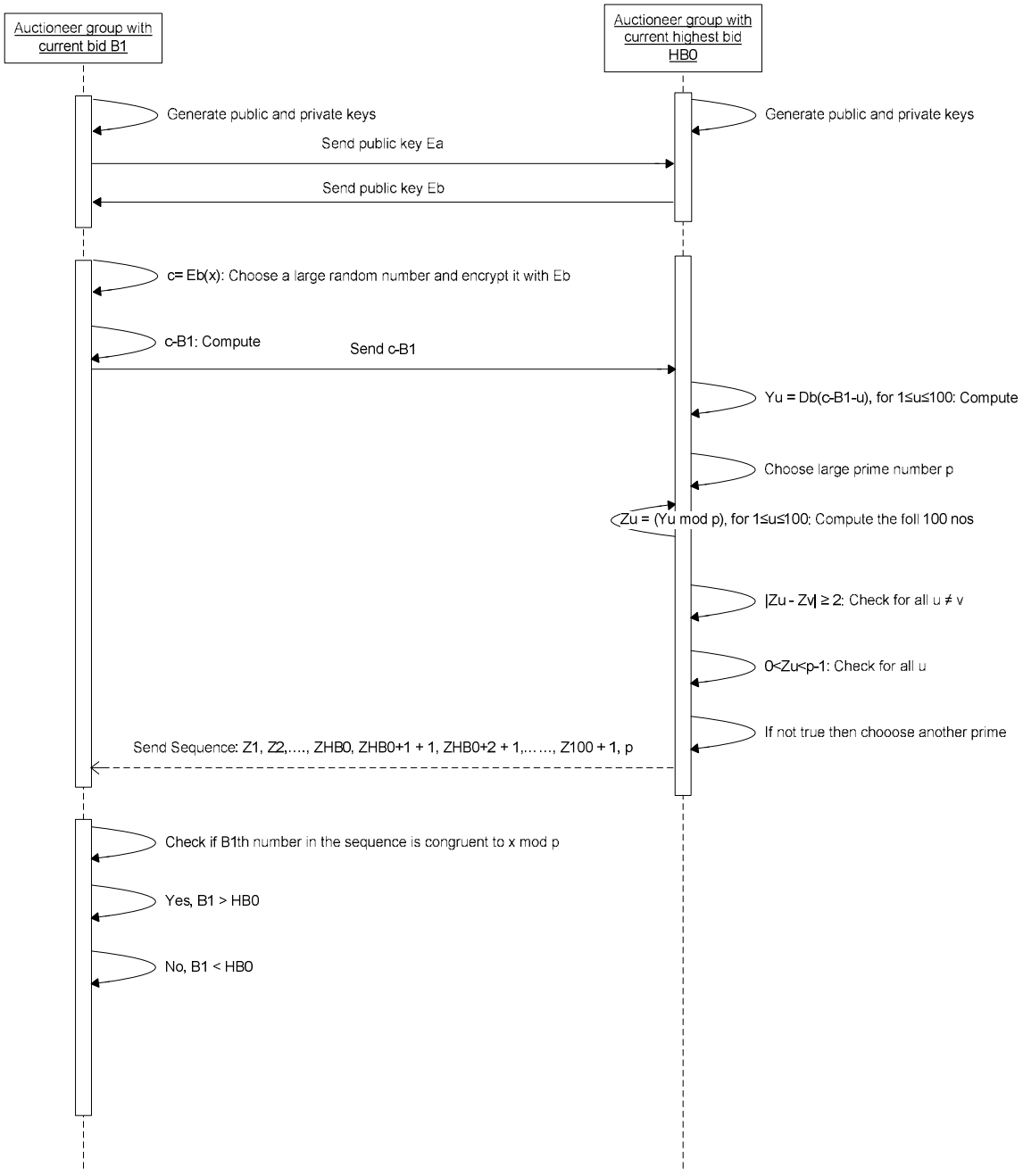

Figure 4.2 Yao Oblivious transfer

6. Group B then verifies that, for all $u \neq v$

$$| Z_u - Z_v | \geq 2$$

and that for all u

$$0 < Z_u < p-1$$

If this is not true then group B chooses another prime and tries again.

7. Group B sends group A this sequence of numbers in this exact order:

$$Z_1, Z_2, \ldots, Z_{HB0}, Z_{HB0+1} + 1, Z_{HB0+2} + 1, \ldots\ldots, Z_{100} + 1, p$$

8. Group A checks whether the B_1^{th} number in the sequence is congruent to x mod p. If it is then group A concludes that $B_1 > HB_0$, else $B_1 \leq HB_0$.

9. Group A tells group B the conclusion.

Now it is assumed that group A will send the wrong result to the group B. But, the creation of bid chains helps in tracing back the group that has cheated. Hence, at any point in the auction only the second highest bid is known to all the auctioneers. The Yao transfer ensures that the bidder at each point submits his true valuation and hence extends the positives of the sealed bid 2^{nd} price auction to an asynchronous, iterative auction.

4.4 Security Analysis

The proposed auction protocol fulfills all the requirements for the ideal auction protocol (see section 2.5) with respect to the auctioneer and the bidding process as it's the same as the protocol proposed by Rolli et al [17]. But we need to see the security provided by the protocol for the secure transfer and comparison of bids. Since, there is no exchange of bid amounts from one auctioneer group to another the highest bid amount is not known to the

any auctioneer group except the group that receives it. If the result is not truthfully revealed by any auctioneer group it can be easily traced back using the bid chain. Also, the chances of an auctioneer group not willing to do the Yao transfer is eliminated since the auctioneer group has different auctioneers and the Yao comparison can be done with the presence of only one faithful auctioneer. Also, the intermediate calculations done during the Yao transfer can be verified by the other auctioneers in the group to ensure that the integrity of the Yao Oblivious transfer.

CHAPTER FIVE
IMPLEMENTATION AND RESULTS

This chapter deals with the implementation of the proposed protocol. It gives an overview of the architecture that is employed for the implementation and gives an insight into JXTA pipes and peergroups and their relevance to the project. It also explains the various classes used in the project and gives pseudo code for the implementation.

5.1 Architecture

The implementation of the protocol is an extension of the implementation realized by Rolli et al [17]. Its architecture can be seen in Figure 5.1.

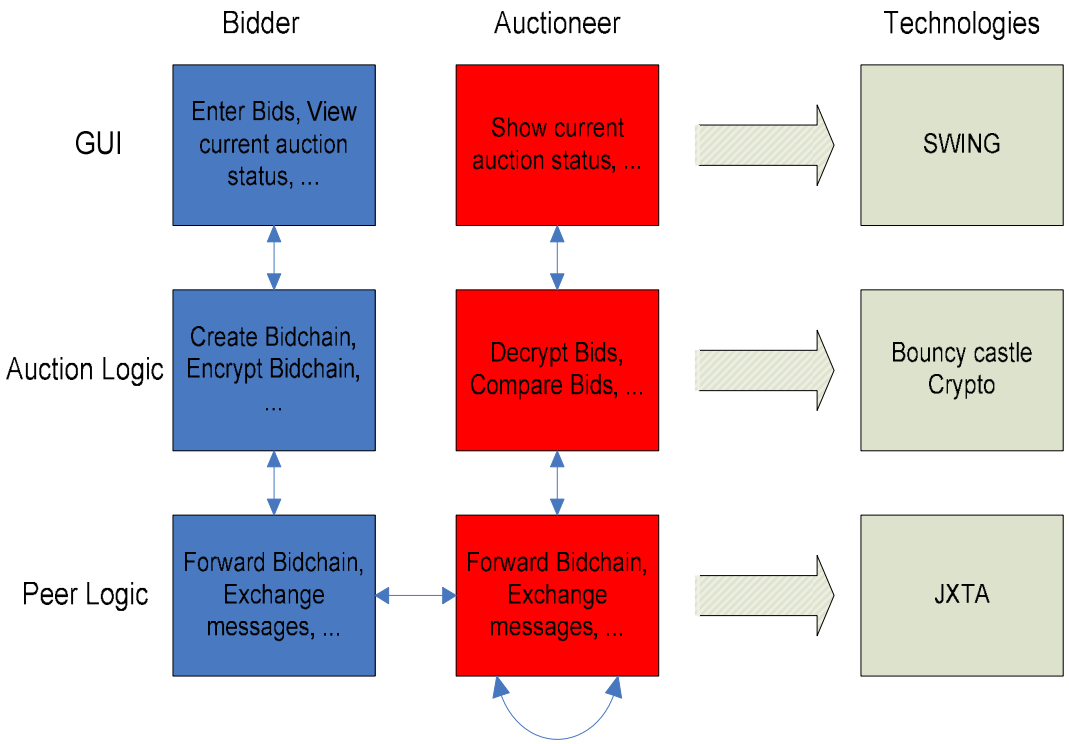

Figure 5.1 Architecture [17]

The implementation is built upon the JXTA 2.3.5 platform. JXTA or Juxtapose [10] is an open source technology created by Sun Microsystems in 2001 and is used to create peer-to-peer (P2P) applications based on Java technology. The various peer nodes i.e. bidder and auctioneer peers, participating in the auction communicate with each other using the underlying protocols of the JXTA framework. Hence, the important functions of the auction, like forwarding bid chains and exchanging messages and auction information is done by the peer nodes using the JXTA framework.

The auction logic is the part that specifies what actions the auction peers should perform based on the requirements of the auction protocol. In particular, the bidder logic creates bid chains and deals with the encryption and signing of the bids. This is done with the help of the routines provided by the Bouncy Castle Crypto API [2]. The auctioneer logic is responsible for decrypting the bids that the auctioneer peer receives from the bidder peer. It also uses the Bouncy Castle Crypto API for this purpose.

The implementation also provides a bidder and auctioneer GUI which uses Swing as the standard Java GUI framework. The bidder GUI takes in the bids from the bidder that will be forwarded to the auction logic module. It also gives a view of the current auction status which includes the current number of items sold, unsold items and the current minimum bid. The auctioneer GUI on the other hand shows the current standing highest bid and the other bids that are lower than it.

5.2 JXTA tools required for Yao oblivious transfer

When an auctioneer peer receives a bid that is greater than the standing highest bid it needs to compare the bid with the peer that holds the current highest bid. This comparison needs to be done in a secure manner using Yao oblivious transfer as described in section

4.2.5.2. We need some mechanism to perform the Yao oblivious transfer between two peers in the auction. This is done using the JXTA pipe service.

5.2.1 JXTA Pipe Service [9]

Pipes are the core mechanism for exchanging messages between two JXTA peers within a JXTA application. JXTA uses input pipes and output pipes for this purpose. These pipes provide a unidirectional, asynchronous and virtual communication channel between two peers.

The peer that wants to initiate communication sets up an input pipe and binds it to a specific pipe advertisement. The peer then publishes the advertisement so that all the other peers that want to communicate with this peer can obtain the advertisement and create output pipes such that messages can be sent to the input pipe. Hence, the input pipe is referred as "receiving" pipe and the output pipe is referred as "sending" pipe. The pipes can function irrespective of the location of the peer, the presence of a firewall or NAT or the network to which the peer belongs to. Hence, if one peer belongs to a TCP network and another peer belongs to a token ring network, they can still communicate provided that JXTA relay peers exist between them.

Pipes support two different types of communication modes namely point-to-point mode and propagate mode.

5.2.1.1 Point-to-Point Mode

The point-to-point pipe connects exactly two pipe endpoints. The output pipe at one end sends messages and the input pipe at the other end receives messages. No reply message or acknowledgement message is supported for this mode.

5.2.1.2 Propagate Mode

The propagate mode connects one output pipe to multiple input pipes. Hence, it is a kind of a broadcast mode in which one peer sends a single message to multiple peers. It might create multiple copies of the same message to achieve the broadcast.

5.2.1.3 Pipes and Peergroups

Pipes and the connectivity of pipes is related to the concept of peergroups. The two peers that want to communicate should belong to the same peergroup. Now in JXTA it is possible for a peer to be part of two peergroups, where one peergroup presides over the other as shown in Figure 5.2.

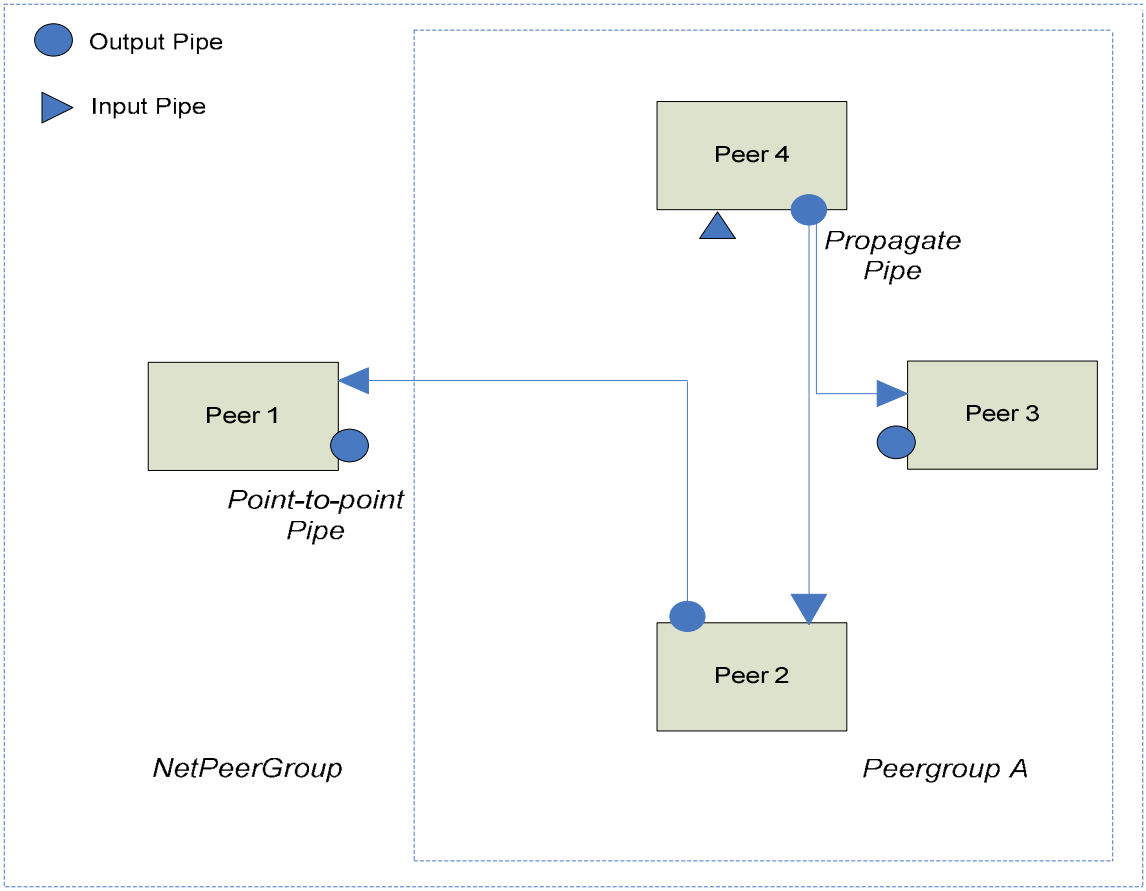

Figure 5.2 Pipes and peergroups

Now, Peer 1 can either use its point-to-point pipe from the NetPeerGroup or the propagate pipes of peergroup A to communicate with peers 2 and 3. Peers 2 and 4 can communicate using the peergroup A pipes, while peer 4 can use the broadcast propagate pipe to communicate with peers 2 and 3. A peer can maintain different pipe connections with the same peer. Hence, Peer 2 can communicate with Peer 1 using both the NetPeerGroup context and the peergroup A context. The only difference is that it will inherit the security settings for the particular peergroup while transferring messages.

In the protocol implementation we have the formation of various auctioneer groups with two or more auctioneers. Each auctioneer within an auctioneer group needs to communicate with every other auctioneer to exchange information like keys, bids, etc. This is possible using the propagate pipes which provide one to many communication and point to point pipes which provide one to one communication. Also, two auctioneer groups need to communicate in order to do Yao oblivious transfer. This is done using the point to point pipe. The NetPeerGroup is the peergroup that is used to do this.

5.3 Yao Oblivious Transfer Implementation

The algorithm used for the Yao oblivious transfer between two auctioneer groups is shown in section 4.2.5.2. The various classes that help in implementing the Yao oblivious transfer are shown below.

5.3.1 Class AuctionStatus

The auction protocol implements a class AuctionStatus that gives a view of the various working parameters of the auction at any instant. The important parameters in the class are shown below:

BidStep standingHighestBid : The standingHighestBid records the second highest bid of the auction. If propagated properly it shows the current second highest bid of the auction.

int HighestBidHoldingAuctioneerGroupNo : The HighestBidHoldingAuctioneerGroupNo records the group number of the auctioneer group that has the current highest bid. Hence, if an auctioneer group receives a bid that is greater than the standing highest bid it looks up at the current Auctionstatus for the group number of the highest bid holding auctioneer and initiates a Yao oblivious transfer between them to determine if the current bid is greater than the current highest bid.

5.3.2 Class AuctioneerpeerImpl

The AuctioneerpeerImpl class implements the JXTA level functionality for the Yao oblivious transfer. It basically implements all the code required for a successful Yao transfer using JXTA pipes. This class initializes the input pipe and the output pipe that will be used for communication with other peers. It then initializes a peerListener which is used to track the changes and progress happening at the upper auction protocol level.It also initializes a PipeMsgListener which is explained in the next section.

5.3.2.1 Interface PipeMsgListener

The PipeMsgListener interface is the container for Pipe Message events. It is used to listen for messages at the JXTA pipe level of the auctioneer peer or peer group. Whenever there is a message event on the input pipe of the auctioneer peer, the PipeMsgListener captures the message and extracts the data that has been sent. But for the message to be extracted the receiving peer needs to know either the string name or the messageID of the function used to send the message.

5.3.3 Class AuctioneerLogicImplYao

The AuctioneerLogicImplYao class implements the major part of the auctioneer logic of the protocol. The bidchains that are sent to the auctioneer group are decrypted and verified. The bid in the bidchain is then compared with the standingHighestBid. If it is greater than the standingHighestbid then the Yao comparison is done, else the standingHighestbid is modified and the auction status is broadcast to all the participating auctioneer groups. This scenario is illustrated in the pseudo code in Figure 5.3:

```
For every newbid that is received
if (newBid.price > doc.startingPrice)
if (newBid > standingHigestBid) {
   get the current auctionstatus
   get auctionstatus.HighestBidHoldingAuctioneerGroupNo
   Do the Yao transfer with the highest bid holding auctioneer group
   if (Yao Transfer.newbidisgreater){
         update this.HighestBid
         update auctionstatus.HighestBidHoldingAuctioneerGroupNo
   } else {
         update auctionstatus.standingHighestBid = newBid
         broadcast auctionstatus to all auctioneer groups
   }
}
At the HighestBidHoldingAuctioneerGroup end
if (Yao Transfer.highestBidhaschanged) {
   send this.HighestBid to the bidchallenging
}
After the transfer and update
```

```
update auctionstatus.standingHighestBid = this.HighestBid
broadcast auctionstatus to all auctioneer groups
```

Figure 5.3 Pseudo code in class AuctioneerLogicImplYao

5.4 Results

The major addition to the proposed auction protocol is the Yao oblivious transfer and 2nd highest intermediate bid mechanism. Since the Yao oblivious transfer contains exchange of keys and transfer of information over JXTA pipes, the exchange time is quite high. But the introduction of the 2nd highest intermediate bid mechanism will try and decrease the amount of time taken for the bid to be processed. In this section we would like to compare the precision of the proposed protocol with the Rolli [17] protocol. Fig 5.4 shows a graph that compares the variation of bid steps for both the protocols with the time taken for the bid step to be created and processed. The bid step for Rolli [17] protocol is increased for every iteration while the bid step for the proposed protocol is kept the same. Thus, the comparison gives us the point where the two protocols intersect. Since, the time taken for bid submission for the proposed protocol is constant at a particular step size it is more efficient for higher bid resolution as compared to increasing number of bid steps in the Rolli protocol. As seen in the Fig 5.3 the protocols intersect at a step ratio of about 4:1.

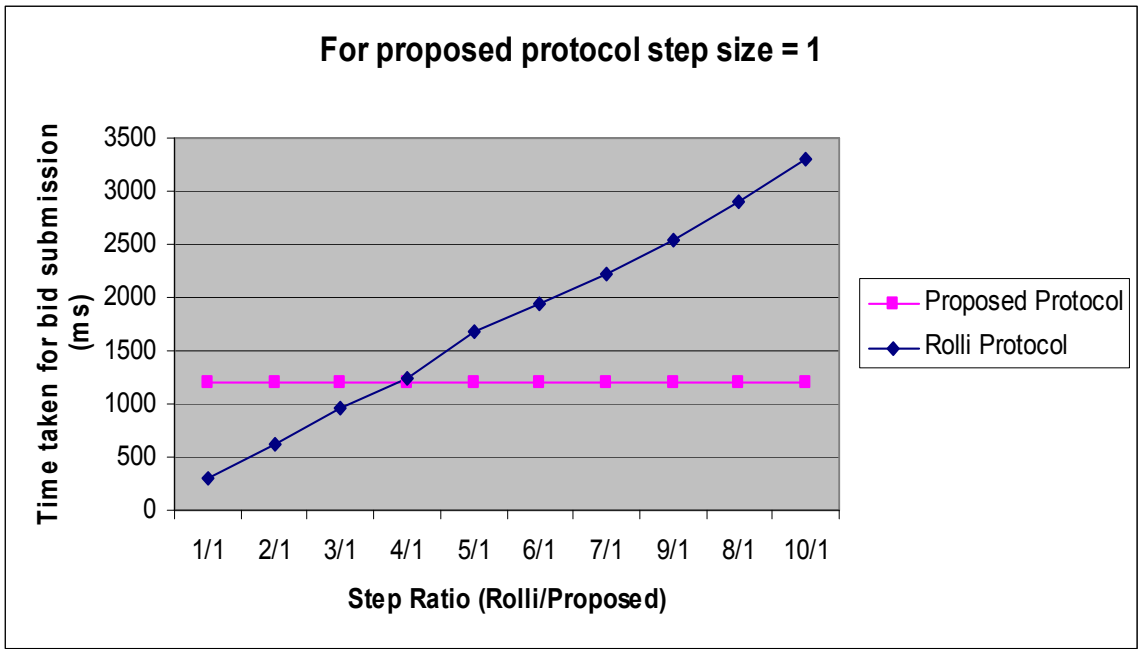

Figure 5.4 Intersection point for proposed protocol step size = 1

Similarly, Figure 5.5 shows the time taken versus the step size for the proposed protocol step size = 2. Hence, for this step size the two protocols intersect at a step ratio greater than 7/2 and less than 8/2. Hence, it is proved that the proposed protocol takes fewer steps to converge to a result than the Rolli [17] method and hence takes fewer iterations.

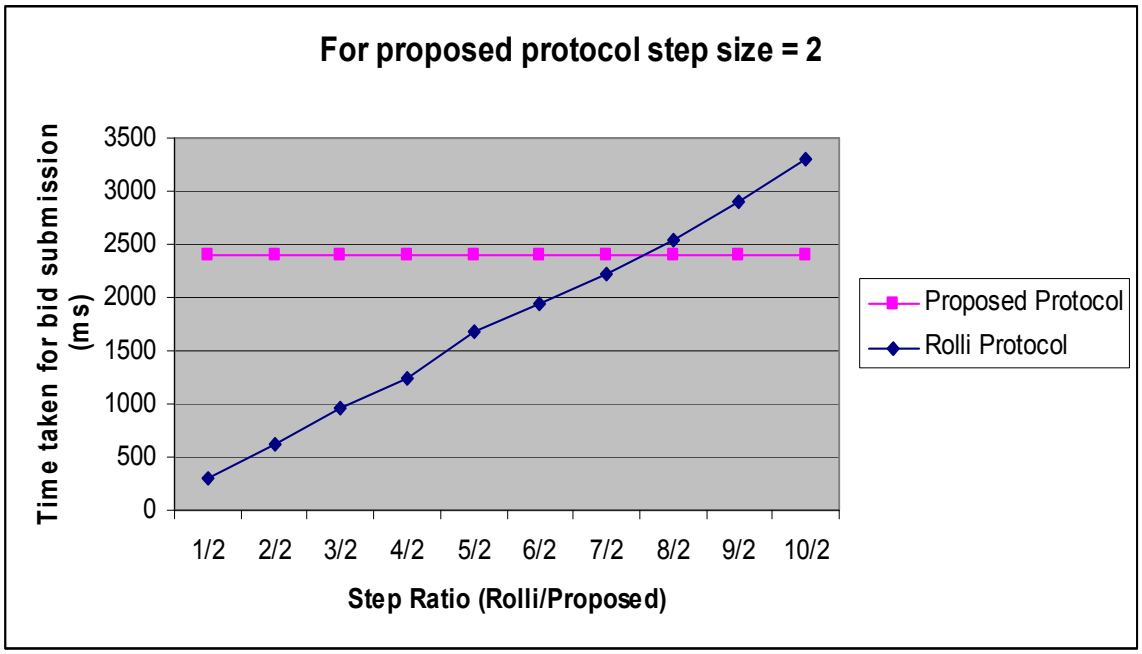

Figure 5.5 Intersection point for proposed protocol step size = 2

5.5 Conclusions and future work

Hence, an implementation of an asynchronous peer to peer auction protocol with Yao oblivious transfer has been realized. The auction protocol has the major properties of being iterative, asynchronous and 2nd price. Apart from these properties it is robust, secure and exhibits a proxy bidding mechanism. The proposed protocol performs better at smaller bid steps than the Rolli [17] protocol. Since, the implementation uses JXTA pipes for the Yao oblivious transfer, the bid comparison rates are higher. But, this is mainly due to the slow nature of the JXTA pipes. If these bid comparison rates are improved the bid precision properties of the protocol can be improved.

The possible future work on this protocol can be its implementation using pipes that are fast, secure and multidirectional which still have to be implemented in JXTA.

REFERENCES

[1] D. Boneh and M. Franklin. Efficient generation of shared RSA keys. *Lecture Notes in Computer Science*, 1294:425+, 1997.

[2] The Bouncy Castle Crypto Package, http://www.bouncycastle.org./

[3] Brandt, F.: A verifiable, bidder-resolved auction protocol. In *AAMASWorkshop on Deception, Fraud and Trust in Agent Societies*, 2002.

[4] Brandt, F.: Fully private auctions in a constant number of rounds. In *7th International Conference on Financial Cryptography (FC)*, 2003.

[5] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *6th ACM Conference on Computer and Communications Security*, pages 120–127, 1999.

[6] Ebay. Electronic auction platform. http://www.ebay.com/.

[7] Matthew K. Franklin and Michael K. Reiter. The Design and Implementation of a Secure Auction Service. *In IEEE Transactions on Software Engineering*, 22(5), pp. 302-312, 1996.

[8] The Gnutella protocol specification v4.0.
http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[9] JXTA v2.3.x Java Programmer's Guide.
www.jxta.org/docs/JxtaProgGuide_v2.3.pdf

[10] The JXTA community webpage. https://jxta.dev.java.net/

[11] S. Katti, D. Katabi, and K. Puchala. Slicing the Onion: Anonymous Routing without PKI. In *HotNets IV*, Nov. 2005

[12] KaZaA Homepage. http://www.kazaa.com

[13] H. Kikuchi. Resolving winner and winning bid without revealing privacy of bids. In *Proceedings of the International Workshop on Next Generation Internet (NGITA)*, pages 307–312, 2000.

[14] M. Naor and B. Pinkas. Efficient Oblivious Transfer Protocols. In *Proceedings of the 12th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 448—457, 2001.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *ACM SIGCOMM Conference*, pages 161–172, San Diego, 2001.

[16] M. S. Robinson. Collusion and the choice of auction. In *RAND Journal of Economics*, 16:141–145, 1985.

[17] Daniel Rolli, Michael Conrad, Dirk Neumannn, Christoph Sorge: "An Asynchronous and Secure Ascending Peer-to-Peer Auction". In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems,* pp. 105-110, ACM Press, New York, 2005.

[18] M. H. Rothkopf, T. J. Teisberg, and E. P. Kahn. Why are vickrey auctions rare? In *Journal of Political Economy*, 98(1):94–109, 1990.

[19] Stajano, F., and Anderson, R. J. The cocaine auction protocol: On the power of anonymous broadcast. In *Information Hiding Workshop (1999), Springer Verlag, LNCS* 1768, pp. 434–447.

[20] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM01*, San Diego, California, USA, 2001.

[21] W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. In *Journal of Finance*, 16(1):8–37, 1961.

[22] Andrew C. Yao "Protocols for Secure Computations". In *Proceedings of Twenty-third IEEE Symposium on Foundations of Computer Science*, Chicago, Illinois, November 1982, 160-164.