

AUTOMATA BASED AIP IN MAS USING FORMAL METHODS

T06672



Developed by

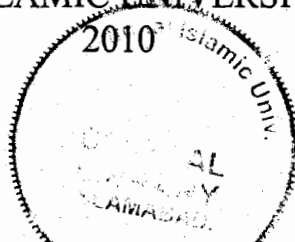
Munira Ghulam Nabi

99-FAS/MSSE

Supervised by

Dr. Aamer Nadeem

DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF BASIC & APPLIED SCIENCES,
INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD.



Accession No. FH6672

SH110

2/10
M.D.

MS
006.3
GHA

- 1- Intelligent agents (computer software)
- 2- Agent interaction protocol (AIP)

~~D-E~~
AC
20.12.10

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*In the Name of Allah The Most Beneficent
The Most Merciful*

Department of Computer Science
International Islamic University, Islamabad

Date: _____

FINAL APPROVAL

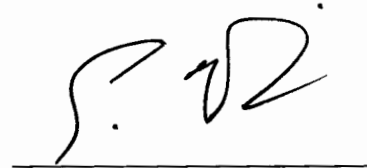
It is certified that we have read the project titled "Automata Based AIP in MAS using Formal Methods" submitted by **Miss Munira Ghulam Nabi Reg. No. 99-MSSE**. It is our judgment that this project is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree MS in Software Engineering.

COMMITTEE

External Examiner:

Dr. Sajjad Mohsin

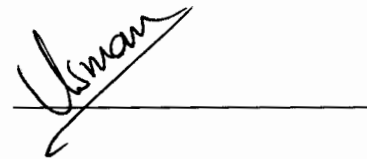
Associate Professor, Department of Computer Sciences,
Comsats, Islamabad.



Internal Examiner:

Mr. Usman Nasir

Department of Computer Sciences,
International Islamic University, Islamabad.



Supervisor:

Dr. Aamer Nadeem

Associate Professor, Department of Computer Sciences,
Muhammad Ali Jinnah University, Islamabad.



**A dissertation submitted to the
Department of Computer Science,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
MS in Software Engineering**

To my Loving Parents

“My Lord have Mercy on them (Parents) both as they did care for me when
I was little”

(AL-QURAN 17:24)

DECLARATION

I hereby declare that this project report, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed the project and its report while working individually, and completed the report entirely on the basis of my personal efforts made under the sincere guidance of my Project Supervisor. If any part of this report is proved to be copied out or found to be reported, I shall stand by the consequences. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other University or Institute of learning.

Munira Ghulam Nabi

PROJECT IN BRIEF

Project Title: Automata Based AIP in MAS using Formal Methods

Undertaken By: Munira Ghulam Nabi

Supervised By: Dr. Aamer Nadeem

Starting Date: October 2007

Completion Date: December 2009

Operating Systems: Windows Vista

Tool Used: Microsoft Word, Z/EVES, Python, Microsoft Visio.

System Used: Intel Pentium IV, 2.25 MHz processor

Table of Contents

Chapter No.	Contents	Page No.
1.	Introduction	1
	1.1 Motivation & Related Work.....	2
	1.2 Problem Statement.....	5
	1.3 Proposed Solution	6
	1.4 Thesis Outline	8
2.	Literature Survey.....	9
	2.1 Cheremisinov & Cheremisinova's Approach.....	9
	2.2 Badica and Badica's Approach.....	10
	2.3 Chen and Sadai's Approach	11
	2.4 Dumas, Governatori, Hofstede and Oaks's Approach.....	12
	2.5 Cheng's Approach.....	13
	2.7 Description of analysis parameters.....	13
	2.8 Conclusion.....	19
3.	Abstract Model.....	21
	3.1 Work Flow of the Proposed Methodology.....	21
	3.2 A Finite State Model for the Proposed Scheme.....	22
	3.2.1 States.....	23
	3.2.2 Transitions.....	24
	3.2.3 Transition Table.....	26
	3.3 Making the FSM Deterministic.....	29
	3.3.1 Conversion of NFA to DFA.....	29
	3.3.2 Derived States.....	35
	3.3.3 Transition Table for DFA of English Auction Protocol.....	36
	3.4 States Minimization.....	36
	3.4.1 State Invariants.....	38
4.	Formal Model	40
	4.1 Global declarations and definitions.....	40
	4.1.1 Char.....	40
	4.1.2 Text.....	40
	4.1.3 Integers.....	40
	4.1.4 Power.....	41

4.1.5 Compute Public Key.....	41
4.1.6 Encrypt.....	41
4.1.7 Maximum.....	42
4.2 Static Formal Model.....	42
4.2.1 Bidder.....	42
4.2.2 Bidder's Private Key.....	43
4.2.3 Round Key Board.....	43
4.2.4 Auction Manager Board.....	44
4.2.5 Registration Board.....	45
4.2.6 Item.....	46
4.2.7 Bidding Board.....	47
4.2.8 Winner Announcement Board.....	48
4.2.9 Access of Round Keys for Auction Manager.....	49
4.3 Dynamic Formal Model.....	50
4.3.1 Display Bidder.....	50
4.3.2 Display Round Key Board.....	51
4.3.3 Display Auction Manager's Board.....	52
4.3.4 Generate Round Key.....	53
4.3.5 Register Bidder.....	54
4.3.6 Generate Auction Ticket.....	55
4.3.7 Display Winner Announcement Board.....	56
4.3.8 Revoke Bidder From Registration Board.....	57
4.3.9 Verify Bidder.....	58
4.3.10 Revocation of Bidder from Round Key Board.....	59
4.3.11 Submit Bids.....	61
5. Evaluation.....	63
6. Analysis of the Proposed Model.....	69
6.1 Analyzing Z specifications.....	69
6.1.1 Error Checking.....	69
6.2 Validation of Formal English Auction Protocol.....	71
7. Conclusion.....	74
References.....	76

List of Figures

<i>Figure No.</i>	<i>Figure Name.</i>	<i>Page No.</i>
Figure 3.1	Work Flow of the Proposed Methodology.....	22
Figure 3.2	NFA of English Auction Protocol.....	28
Figure 3.3	Minimized Deterministic Finite Automata of EAP.....	38
Figure 6.1	Snapshot of Z/EVES Proof Window of GenerateRoundKey schema.....	73
Figure 6.2	Snapshot of Z/EVES Proof Window of RegisterBidder schema.....	73

List of Tables

<i>Table No.</i>	<i>Table Name</i>	<i>Page No.</i>
Table 2.1	Comparison of the properties of English Auction Protocol.....	17
Table 3.1	Invariants abbreviations of EAP.....	23
Table 3.2	State Invariants of NFA of EAP.....	24
Table 3.3	Operations of EAP.....	25
Table 3.4	State transition table for NFA of EAP.....	26
Table 3.5	State transition table for DFA of EAP.....	36
Table 3.6	State transition table for minimized DFA.....	37
Table 5.1	Comparison of the properties of EAP.....	65
Table 6.1	Results of exploration of Static Model of EAP.....	71
Table 6.2	Results of exploration of Dynamic Model of EAP.....	72

Abstract

Multi-agent system (MAS) has emerged as one of the most practical solutions with built in flexible interactions among various agents, for analysis, modeling and design of a system. For successful working of autonomous agents in complex systems like MAS have to follow Agent Interaction Protocol (AIP). Since, interaction among agents requires a set of agreed messages and rules for taking appropriate actions according to the nature and assumptions of the communication channels. As MAS depends a lot on the significant role of AIP hence, its formal specification and verification is substantial. Use of secure interaction protocols is emerging with the ever increasing popularity of public internet auctions. This work aims to integrate automata and Z to comprehend and represent multi-agent interaction systems. English Auction Protocol (EAP) has been taken as an example. The proposed approach groups and formalizes various security properties like anonymity, traceability, no framing, linkability in a round of auction, unlinkability in different rounds of auction, unforgeability, fairness, one time registration, easy revocation and verification among agents; while taking advantage of behavior handling and semantic characterization of automata. A formal model of EAP using automata is presented to ensure validation of security properties to achieve reliable and secure EAP.

1

Introduction



1. INTRODUCTION

Multi Agent System (MAS) is a technology that has been developed since early 1990's. It is the most practical solution for analysis, modeling and design of any system, where there are several autonomous agents working together and have distributed control amongst them (Busoniu et al., 2005). Another vital feature that makes MAS unique from other systems is their built-in ability of flexible interactions (Brazier et al., 1997).

However, due to the autonomous nature of agents, the design and development of MAS is more difficult as compared to single agent systems. MAS depends on the significant role of Agent Interaction Protocol (AIP) (Brazier et al., 1997), its formal specification and verification (Chainbi et al., 1998). For the successful working of the agents, AIP applies different rules to manage and coordinate the communication among different agents. Due to the complex nature of interaction systems, Littman et al. (2002) proposed that it is difficult to verify, validate and reuse them. Especially the concurrency, reactivity, autonomy, openness and extensibility of MAS bring new challenges for its implementation (Busoniu et al., 2005).

Interactions of agents require a set of agreed messages and rules for actions that must be applied on the incoming messages according to their nature and assumptions of the communication channels. Although agents are anonymous in nature, they still have to follow the interaction rules. Therefore, agents can be predictable because they follow interaction rules (Ambroszkiewicz et al., 1999).

Protocol representation for mutual conventions in formal language is necessary (Cheremisinov & Cheremisinova, 2006). Indeed, each formalism have constructs which enable complex structure specification. Moreover, aspects such as reactivity and concurrency can be easily dealt with (Hilaire et al., 2000).

Agents involved in non concurrent interaction protocols for conversation are usually specified as Deterministic Finite Automata (DFA), various examples are shown in literature (Cheremisinov & Cheremisinova, 2006). In interaction protocols, the transitions

of DFA specify the communicative measures among agents in a conversation. A DFA based protocol representation corresponds to a category of well-determined conversations.

Certain alternative approaches can be used for the representation of Agent Interaction Protocols, such as AUML or state charts, Petri nets. However, representing protocols of realistic complexity using state charts or AUML requires considerable development and debugging labor (Cheremisinov & Cheremisinova, 2006). Standard Petri nets fail to express the alternative actions and states. Hence, the composition of Z and FSA seems particularly suited in order to specify MAS.

The behavior of the system can be defined in a very well manner using FSA and the transformation of FSA to formal model is easy. Formal methods can help increase the correctness and trustworthiness of the software developed (Kneuper, 1997). The use of a formal specification language such as Z ensures the correctness, reliability and consistency at analysis and design stage, before the actual implementation of the software system starts.

1.1 MOTIVATION AND RELATED WORK

With the ever increasing popularity of public internet auctions e.g. eBay and Yahoo auctions, its dynamic behaviors are becoming more popular now a days. It is, therefore, necessary to conduct a comprehensive study to identify and furnish this feature. Electronic auction is an attractive form of electronic commerce and recently many kinds of auction services are provided over the internet e.g. e-Bay, founded in 1995, has grown into an online auction house with 135 million registered users (Roggero et al., 2005). Electronic auctions can be categorized into sealed bid and public auctions (Lee et al., 2001). In a sealed bid auction, bid secrecy is the main concern. Each bidder secretly submits a bid only once in a bidding stage of a sealed bid auction. At last, the bidder who offers the highest price is announced as a winner. In public auction, anonymity is the major concern. All the bid values are published but the bidder participates in an auction

protocol anonymously. Complex activities like auctions that recurrently come across in e-commerce practices are utilized through strong interaction of business processes. The study and development of formal modeling is needed for the automation of auctions using agents. This provides a strong motivation for the formalization of these complex systems (Cheng, 2006). The well known types of auctions include English auction, Dutch auction, Sealed first-price auction, and Sealed second-price auction. Many researchers are in a favor of Vickery auction by saying that it is more suitable in online auctions because of its shilling free nature. The sealed bid and bid only once characteristics of Vickery auction make them more effective. However, the competition principle does not work well in this type of auction and bidders cannot be able to make their own evaluations. Hence, the complicated and rigorous problems occur due to the shilling but online auction houses still prefer English auction (Cheng, 2006; Roggero et al., 2005).

Foundation for Intelligent Physical Agents (FIPA) has standardized the agent interaction protocol for EAP (FIPA, 2000), which has been taken as an example in this thesis. This functional standard of EAP has been used with certain extensions made to enhance its significance in internet auctions. In English Auction all the bid values are published in an anonymous way. Each bidder offers the higher price one by one and can bid multiple times. This public auction scheme is considering the Registration Manager and Auction Manager to execute the randomization operation in round setup process. It makes the identity of the winner to be published while keeping the anonymity of the winner in preceding rounds of the auction. Auction Manager provides the ticket identifier which is recognized by the bidder only. The two Managers i.e. Registration Manager and Auction Manager perform different operations. Registration Manager is the incharge of one time registration process and participates in the Round key setup process to publish round keys for the bidders and also publishes the winner specific information on his bulletin board in the Winner Announcement board. Registration Manager publishes the identities and public keys of the registered bidders on the Registration board. Also, the Round keys computed by the Registration Manager are displayed on the Registration board. Auction Manager prepares auction tickets in each round of auction using a random number and round keys, also has some winner specific information to be displayed on the Winner

Announcement Board. Auction manager computes the auction tickets for the valid bidders listed in the Round key board. Bidder has to initially register to the Registration Manager to participate in auction using auction ticket, bidder has got his private and public keys and only the higher bids than previous highest one can be posted on the Bidding Board by bidder. Finally, the bidder with the highest bid is decided from the Winner Announcement Board (Lee et al., 2001).

Related work in the literature shows that the existing formalizations of public auction protocols like English Auction Protocol do not incorporate the security features like anonymity, traceability, unforgeability, unlinkability among two different rounds of auction, linkability in one round of auction, revocation, no framing and one-time registration (Badica & Badica, 2008; Cheremisinov & Cheremisinova, 2006, Chen et al., 2003; Dumas et al., 2002; FIPA, 2000). If these security characteristics are not incorporated then the typical problems faced in internet auctions include Bid shielding (Boyd & Mao, 2000) in which a high value bid is withdrawn at the last minute allowing a low bid to be accepted, and Bid siphoning (Boyd & Mao, 2000) in which a seller of an auction can directly contact a bidder and offer an alternative available item. This allows sellers to obtain buyers without paying commission to the auction site. Other problems include Shilling (Cheng, 2006) i.e., introducing fake bids in order to drive up the price of the bid, Snipping (Boyd & Mao, 2000) i.e. is last minute bidding in order to prevent other bidders from responding, escrow services problem, and fairness in case of winner etc. The literature shows the importance of such features for secure public auction protocol over the internet like that of EAP (Omote & Miyaji, 2001; Chaum et al., 1991; Chen, 2004; Lee et al., 2001). The reliability of a system can be ensured using two ways, i.e. one is through exhaustive and thorough testing and the other is the use of formal methods. For very small systems it is almost impossible to do exhaustive testing. There is no 100% guarantee on the reliability of a system but formal methods gives surety of correctness that is why it is always used in critical systems like air crafts and nuclear power plants.

It is appropriate to use formal methods to specify the properties of an auction interaction protocol, as it is not trivial and it ensures the accurate working of previously mentioned security properties. Since, MAS depends a lot on AIP (Brazier, 1997) and it's formal specification. Therefore, we explore the applicability of formal model oriented approach Z in multi-agent systems for the specification and verification of the properties for interaction protocols and simulation of their behavior before implementation.

1.2 PROBLEM STATEMENT

Existing literature shows that many significant properties of AIP have not been formalized. Different pieces of work have catered to a different set of properties like for instance, Cheremisinov and Cheremisinova (2006) formalize concurrent behavior, but security properties have not been considered. Only deadline property has been considered by Pitt et al. (2000). Formalization of properties like deadline, validation and verifiability have been considered by Dumas et al. (2002) and Cheng (2006).

It has been observed from the existing literature on formalization of English Auction Protocol that major properties necessary for security are not paid much attention. Properties like anonymity, traceability, no framing, one-time registration, easy revocation, verification, unforgeability, linkability within a round of auction, unlinkability among different rounds of auction and formal analysis ensure efficient, secure and conveniently implementable English Auction Protocol. Some of these security properties have been taken into consideration by the existing techniques for formalization of EAP. Dumas et al. (2001) formalized the traceability, validation and deadline only, Chen et al. (2003) only paid attention to the formalization of easy revocation, verifiability, validation and deadline, where Cheng (2006) only considered traceability, verifiability and validation, Cheremisinov and Cheremisinova (2006) formalized easy revocation, one time registration, validation and dead line, and Badica and Badica (2008) provides us with the formalization of easy revocation, verifiability and validation only. All these techniques in literature have not paid attention in formalizing important security properties of EAP like anonymity, no framing, unlinkability in different rounds of

auction, linkability in a round of auction, Unforgeability, and fairness. It is obvious from literature that the lack of security properties cause problems like Bid shilling, Bid shielding, Bid siphoning etc. (Boyd & Mao, 2000). Considering the importance of these unattended security properties, it is important to combine all these properties at one platform to get more reliable, effective and secure EAP formal model.

According to the US internet fraud report in electronic auctions, approximately 87 % of sites face a security threat (Boyd & Mao, 2000). Ambiguities and inconsistencies remain even if security properties are brought in but not formalized.

Considering the security properties in agent protocols are not trivial, it is suitable to ensure their correctness using formal methods which can specify and verify the properties of English auction protocol, and simulate their behavior before implementation. By including the security properties only, we cannot prove that a system is error free. It is then necessary to use more powerful techniques of automated analysis such as verification. Indeed, the agents and their communication should satisfy some important security properties such as anonymity, fairness and privacy (Boyd & Mao, 2000). The lack of the formalization of security properties leaves the ambiguities and inconsistencies in the English Auction Protocol, leaving a loop hole that propagates during implementation phase. The mathematical structures provided by Z allow us to write service specifications that are abstract, independent of implementation detail, and closer to the user's point-of-view of the service (Airchinnigh & Butler, 1993).

1.3 PROPOSED SOLUTION

A holistic approach has been proposed that formalizes the EAP by adding non-trivial properties that are necessary for secure EAP using Finite State Automata (FSA). The EAP has been taken in account for the sake of proving the true conversation (Munira, N., & Nadeem, 2009). For a fixed structure representation of such conversations defined in this way, some kind of graphical representation is needed (Cheremisinov & Cheremisinova, 2006). The properties that are considered from Lee et al. (2001) are

anonymity, traceability, no framing, one-time registration, easy revocation, verification, unforgeability, linkability in a round of auction, unlinkability among different rounds of auction and formal analysis. The deterministic finite state automata has been initially designed to gain the modeling power of the system as visual flows, states and transition functions can be used to manage automata for system control behaviors. After that Z specification language has been used for more refined and detailed system model of EAP. This formal model has been verified using Z/EVES tool set (Meisels, 1997).

Many formal specification languages exist in the literature, e.g. Z (Spivey, 1992), Pralu (Cheremisinov & Cheremisinova, 2006), Prolog (Chen et al., 2003), etc. However, the focus of this research is Z specification language since it is one of the most widely used formal specification languages in the industry (Huaikou, Ling, Chuanjiang, Jijun, & Li, 2001). Also it has been widely used in formalizing many safety critical systems such as Air craft system, Nuclear Power plants etc. It is supported by a number of automated tools (Bowen, 2006). Z specification language was initially proposed in the late 1970s but became mature in 1980s. It was largely developed by the Programming Research Group (PRG) at the Oxford University Computing Laboratory (OUCL). Later, it was standardized by ISO (International Standards Organization) (ISO/IEC, 2002; ISO/IEC, 2007).

The use of FSA provides us with possible states and operation sequences of EAP. These states and operations are then formalized in to Z states and operation schemas respectively.

The identified states and operation sequences of EAP obtained from FSA and their formalization in Z helps in achieving agent's negotiation and cooperation. The case study under consideration in this research is English Auction Protocol. A model at an abstract level has been developed as Deterministic Finite Automata (DFA) and then there is a conversion to formal model in Z specification language. This model is finally verified using a tool named Z/EVES toolset (Decker, 1994).

1.4 THESIS OUTLINE

The rest of the thesis is organized as follows. Chapter 2 evaluates the literature survey about the formalization of English Auction protocol. It explains the comparison of existing formal approaches used for English Auction protocol and the role of formal method in software development process. Chapter 3 begins with the actual framework of English Auction system by graphical representations and visual flows by making Deterministic FSM, examining the input and output spaces, components and their relationship to propose an initial abstract model. Chapter 4 presents the formal framework for English Auction protocol using Z notation. Z state schemas are used to formalize system components and relations while operational schemas define the transitions of the English Auction Protocol. Chapter 5 demonstrates the evaluation of the properties of English Auction Protocol. Chapter 6 illustrates the analysis and verification of all the software system specification against a tool i.e. Z/EVES toolset. It is a tool for analyzing Z specifications. It can be used for parsing, type checking, Domain checking, schema expansion, precondition calculation, refinement proofs and theorem proving. Chapter 7 gives the concluding remarks about the project work.

2

Literature Survey

2. LITERATURE SURVEY

This chapter discusses the related work that has been done on interaction protocols. Different research works in multi-agent systems which have focused on English Auction Protocol formally and its significant properties informally are analyzed in depth and described in detail. At the end of chapter, a tabular comparison of related existing techniques is also given.

2.1 Cheremisinov & Cheremisinova, 2006

Cheremisinov et al. (2006) presented parallel logical control algorithms in formal language PRALU for describing AIP. This approach can be used for modeling complex and concurrent conversations among agents in multi-agent system. It can be used to define complex conversations that are composed of great number of simple conversations. For the specification of AIP, an example of EAP has been taken. In this process, Auctioneer starts with sending the message and then other participants will continue communication. The PRALU blocks which are graphically presented in this work are Main process, Auctioneer and Buyer.

This paper describes the use of formal theories for concurrency and distribution in interaction protocols for real time MAS. With the language PRALU, the concurrent characteristics of conversations are expressed graphically, the dynamic state of complex conversation is captured and conversation structure for processing multiple concurrent messages is described.

Remarks

The specifications given in this paper are very abstract, as only two agents, buyer and auctioneer are involved in the whole auction process. To get complete understanding of an auction process there is a need of refinement of these specifications in a detailed description and inclusion of more agents like administrator agent, seller agent and their processing. There is a problem faced by using Petri Nets, that alternative actions either agree or reject, but not both can be expressed in standard Petri Nets.

There are a lot of different non trivial aspects of English Auction Protocol like, what basis are required to made of an increment in bidding price of a bidder. Only the concurrent behavior of EAP has been described. However, the approach does not discuss the security aspects, which are vital for interaction protocol used in public auction services provided over the internet.

2.2 Badica & Badica, 2008

Badica et al. (2008) used finite state process algebra to propose a formal framework for modeling agent interactions in agent based English auctions. A case study with single item in EAP is demonstrated by applying the framework to model agent interactions. In this paper the mapping of local processes indexed with subsets have been taken in consideration. A subset index was mapped to a sequence of integer indexes representing subset elements. There is a mapping of FSP model of English Auction system for three buyers and FSP language is supported by LTSA tool. This tool supports a way of checking a target system using safety and progress properties. There are the stages of English auction presented in the form of FIPA ACL messages. FSP actions are used for modeling of EAP, which are initiation, buyer registration, bid submission and agreement formation. There is a server process that describes the negotiation host role, the buyer and the seller processes and the system process as parallel composition of negotiation host, buyers and seller processes.

The system is initialized by creating two buyer agents and one seller agent. Seller role has some indexed set of actions etc. The work bridges the gap between the analysis of agent interactions using formal methods and the implementation of agent based English auctions using available agent platforms. This approach provides formal verification of system against qualitative properties and it can be used to derive quantitative simulation. Hence, proving that the development of formal models can be used as a basis for the sound implementation of negotiation agents by mapping local processes to agent behaviors.

Remarks

The message contents with the exception of buyer identities are ignored in this model. Focus is only towards the interaction patterns between negotiation participants, and the mapping of server process is not scalable with respect to the number of buyers. This approach lacks the details and formalization regarding bidder's privacy and anonymity.

2.3 Chen and Sadaui, 2003

Chen et al. (2003) deal with the formal specification, simulation and model checking verification of an agent based online auction. The focus is towards the applicability of techniques and tools from distributed systems. Formal specification language such as LOTOS is used for specification and verification of the properties of interaction protocols in multi-agent systems.

EAP has been taken as a case study. A single auction with one to many relationship, like one item by one seller to n buyers has been considered, who submits their bids to the auctioneer. The agents used in auction interaction protocol are Administrator, Auctioneer, Seller and Buyers. The communication channels are unidirectional, as agents can either send or receive messages. Agent can read messages from buffer and also can send message to another agent. Architecture of protocol as well as buy agent process is specified in LOTOS. Auction failure has been taken as an example to validate the conformance of specification to initial requirements. Finally, the correctness verification of interaction protocol based on safety and liveness properties is verified using Evaluator tool.

Remarks

Authors have made use of existing techniques and tools like LOTOS and CADP to design and specify interaction protocols in multi-agent system. Hence, providing a basis for verification and validation of system's functionality. EAP has been taken as a case study, in which the focus is only towards buy agent along with architectural specifications. No specification is given for the rest of resources like Administrator, Sell and Auctioneer agents. LOTOS has some limitations such as it cannot specify quantitative time and

exception handling. However, plural rounds of auction with one time registration are not possible in a verifiable way and consequently have not been formalized as well.

2.4 Dumas et al., 2002

Dumas et al. (2002) emphasized on the negotiating agents, which combines UML state charts with a defeasible logic. The correctness has been validated through examples of negotiation strategies. This approach is applicable for developing agents capable of participating in multiple concurrent negotiations. The internal coordination of the negotiating agents is expressed through UML state charts.

This approach is applied to case study of EAP. A bidder's strategy is expressed using defeasible logic. The conditions of increment in an auction are described, that to remain in an auction what will be the minimum amount that raises the valuation of the user. In case of English auctions, the user will start by bidding the reservation price, and will subsequently overbid the other participants' bids by the minimum increment, as long as the resulting bid is less than the current valuation. However, if the auction's deadline is too close, he will bid his current valuation instead of just overbidding by the minimum increment. The control module of the bidding agent has been explained in which the parameters, status, and history of the auction, are modeled through the predicates, and constants. The defeasible logic programs appearing in the paper have been tested using a defeasible logic inference engine.

Remarks

In this paper, the issue of inter-agent message exchange has not been addressed. For modeling defeasible logic is used, which is more accurate and all the programs are valid that are used in this paper as they are validated by using defeasible logic inference engine. The focus in this work is more towards communication module and reasoning module but does not tackle the security aspects which are very important for open auctions like English auction scheme.

2.5 Cheng, 2006

Cheng (2006) uses model checking techniques in concurrent online auctions for the detection of shilling traits by proposing a formal approach. English auction protocol is being used as a case study. Firstly, the specification language of a model checking tool SPIN is a basis for the development of a model template. Two concurrent biddings can be simulated via this model. LTL formulae can then be applied to confirm normal or shilling behavior in the biddings.

The two major kinds of shilling behaviors are reserve price shilling and competitive shilling. In this approach the model is neither a system nor a protocol. Due to the quantity and complexity of the data, it is impossible as well as inefficient to read and detect bidder's behaviors manually all the time. Combining the model with the specified LTL formula SPIN can be used to generate a corresponding model verifier. Upon execution of the model verifier, the validation of the input given to LTL formula is confirmed.

Remarks

Shilling behavior is a severe problem in online auctions; this approach is one of the attempts to tackle this problem. In agent based auction market, the bidding behavior is more complex since there will be more bids placed by autonomous agents in each auction. The serious problems like shilling behaviors can be improved by adding security features. However this approach does not make such features a part of EAP except for the traceability of the bidder.

2.6 Description of analysis parameters

This segment describes the identified analysis parameters to compare the competence and effectiveness of existing formal properties of English auction protocol. On the basis of these parameters, analysis template is created to give rapid insight on each of the approaches explained as shown in Table 2.1. The parameters are:

- ↓ **Language Used:** "Language used" means that on which formal specification language the respective approach was demonstrated.

✚ **Security Properties:** Only a few studies on English auction have been reported to date. The main reason behind that is the trade-off seen in many types of auction schemes like that on Japanese auctions. This trade-off usually result in compromising security features and focusing more on functional requirements like efficiency. However, it is seen that this compromise results in huge financial loses. According to a report (Wilson, 2008) Yahoo Japan Corporations auction Website has been illegally accessed about 1.5 million times with codes and passwords stolen from members from an IP address in China. The thieves used stolen passwords and a common IP address to break into users' accounts and sell goods fraudulently via Yahoo Japan. Related work in literature shows that the existing formalization of public auction protocols like English Auction Protocol did not pay attention to the security features including anonymity, traceability, unforgeability, unlinkability among different rounds of auction, linkability in a round of auction, revocation, no framing and one time registration. The literature shows the importance of such features for secure public auction protocol over the internet like that of EAP (Chen, 2004; David et al., 2003, Lee et al., 2001; Omote & Miyaji, 2001).

a) **Anonymity:** “Anonymity” means whether anybody can identify a bidder from one’s signature on a bid. Unless Auction Manager and Registration Manager conspire, discriminating the identity of the bidder during the auction is impossible. Auction Ticket is different for every round so the identity of bidder can’t be determined. Registration Manager cannot identify the bidder from auction tickets, because the Round key generated by the Registration Manager uses the public key of the bidder and then Auction Manager does the computation on that Round Key and generates the Auction Ticket, so the correspondence of the bidder’s identity with the Auction Ticket is not known to the Registration Manager. Same is the case with the Auction Manager, as the correspondence between the Auction Ticket and the public key of the bidder has not been recognized by the Auction Manager.

b) **Traceability:** “Traceability” is described as whether a winner can deny

about submitting the winning bid after the winner decision procedure or not. While announcing the winner, auction ticket and the secret id of bidder is posted by Auction Manager and the Round key for that auction key is posted by Registration Manager, anyone can identify and confirm the real identity of the winner.

c) No framing: “No framing” depicts whether any body can impersonate a certain bidder or not, as the private key of the bidder is only known to the bidder.

d) Unlinkability among different Rounds of an auction: This parameter shows whether each auction key is different among plural auctions because the secret values by the Registration Manager and Auction Manager are different in every auction, along with a bid or not. So nobody can link two signatures among plural auctions.

e) Linkability in a Round of an auction: This parameter depicts whether anybody can link the same bidder’s bids and the number of bids placed by the same bidder in an auction from the signature because a bidder uses the same public key equipped with different keys assigned by the Registration Manager and Auction Manager.

f) Fairness: “Fairness” demonstrates whether all the bids are fairly dealt with or not. Fairness in the process can be achieved because the Bulletin boards are used by AM and RM, therefore the behavior of the managers will turn out on the Bulletin Boards and any bidder can point out any misbehavior. Fairness is not precisely defined by Lee et al. (2001) in his work.

g) Unforgeability: “Unforgeability” portrays whether anybody can forge a bid with a valid signature or not. Analyses can be used to confirm that

attackers are unable to calculate a legitimate auction certificate as the bidders participate in auction is only the valid bidders.

h) Easy Revocation: “Easy Revocation” explains withdrawal of a bidder from an auction or if Registration Manager is intended to revoke a certain bidder, a revocation of bidder can be frequently conducted and it must be simple and easy. It is easy to revoke a bidder because Registration Manager has only to delete a bidder from Round key board and the Registration board.

i) Verifiability: “Verifiability” is defined as whether anybody can verify a signature on a bid and can confirm whether the bidder is valid or not, as when winners are announced the auction ticket and round key along with the bidder’s id are displayed so that any one can check out its verifiability.

j) One time Registration: “One time registration” shows whether bidder has to register once for the whole auction procedure or have to re register in every next round.

✚ **Validation:** Formal specifications have been validated using some tool.

✚ **Deadline:** The deadline date has been mentioned and formalized in the specifications or not.

Table 2.1 shows the comparison between different formalization techniques of English auction protocol in Multi Agent systems. The parameter “language used” means that on which formal specification language the respective approach was demonstrated.

The parameters “Deadline”, “Easy revocation”, “Validation”, and “One time registration” have been assigned “Yes” value in Cheremisinov & Cheremisinova (2006) to indicate the presence of formalization of these properties. The approach has added “timeout” feature in the formal specification to apply deadline. “Timeout” operation in the formal specification has been used to capture deadline. This operation means to wait for time out

unit times before doing something followed by it. Revocation of bidder is possible in a way as the approach is using PRALU language which allows concurrent and alternative branching in the formalization of Buyer process after the auction starts. Buyer can whether propose the price by using Price_proposed operation or he is able to quit from the auction using End_auction operation. In this approach, validation of the formal specifications of interaction protocol in multi-agent systems can be done using existing software for PRALU. The hierarchal description of algorithms can be supported by the language PRALU. One time registration is applicable in a way as in “Buyer” process after the timeout operation the loop will again go back to the “start_auction”, from where the buyer can propose the price again. The registered buyers will be entered in the Buyer process and the registration will be in the main process. After registration in the main process the bidders can participate in the multiple rounds of auction.

Table 2.1 Comparison of the Properties of English Auction Protocol

Author	Dumas et al. [2001]	Chen et al. [2003]	Cheng et al. [2006]	Chere misino vet al. [2006]	Badica et al. [2008]
Language used	DeLp	LOTOS	LTL	PRALU	FSP
Security Properties					
a) Anonymity	No	No	No	No	No
b) Traceability	Yes	No	Yes	No	No
c) No framing	No	No	No	No	No
d)Unlinkability in different rounds of auction	No	No	No	No	No
e)Linkability in a round of auction	No	No	No	No	No
f) Fairness	No	No	No	No	No
g) Unforgeability	No	No	No	No	No
h) Easy Revocation	No	Yes	No	Yes	Yes
i) Verifiability	No	Yes	Yes	No	Yes
j) One time registration	No	No	No	Yes	No
Validation	Yes	Yes	Yes	Yes	Yes
Deadline	Yes	Yes	No	Yes	No

Table 2.1 depicts that the parameters “Easy revocation”, “Validation”, “Deadline” and “Verification” have been assigned “Yes” in Chen et al. (2003) to confirm the formal specification of these properties. The approach has incorporated “subscription” and

“unsubscription” features for a bidder at any time during the auction to apply revocation easily. Traceability is achieved in a manner that only the registered buyers can participate in the auction. Auctioneer keeps a check using feature “Accept proposal” of a bidder after checking the validity. Then, auctioneer notifies all the participants and administrator agent about the auction result. Otherwise, auctioneer replies “Reject proposal” to invalid bid and gives the reason why the proposal is rejected. Validation is respected in a manner that the tool EXIBITOR allows generating one or all possible scenarios that satisfies the user defined goals. The interactive simulation of LOTOS specifications allows tracing and monitoring all possible execution sequences and detecting errors. The approach has added the verification in a way that the proposal cannot be submitted by a bidder before subscription. Using formal specifications, the semantics of a system are described precisely, providing a basis for the verification and validation of the functionality of the system.

The parameters “Traceability”, “Deadline” and “Validation” have been assigned “Yes” in Dumas et al. (2002) to represent the existence of formalization. A memory module which contains the history of the past decisions and the interactions of the agent, including the current intentions, realizes the traceability feature. The approach has added “Timing_Constraints” feature in the formal specification to apply deadline. The auction stops when the Timing_Constraint is violated; i.e. either deadline is reached or no bid is registered for longer than established maximum duration and predicate $time_remaining(T)$ gives the remaining time before the end of the auction. The validation is respected in order that all the defeasible logic programs in this approach have been tested using defeasible logic inference engine.

Table 2.1 shows that the parameters “Validation”, “Traceability” and “Verifiability” have been assigned “Yes” in Cheng (2006) for the sake of presence of formalization of these parameters. The parameter Validation is satisfied as using model checking results, we can track the bidder’s bidding history. When the auction model is executed, the model verifier will show whether the result is valid or not. For the realization of Traceability, the pattern based approach is used for the specifications in which each pattern is assigned

with different scopes like Before Q , After Q etc. The two patterns that are used in this approach are Absence pattern and Existence pattern. In Absence pattern scope “Before R ” is described by the formula $\diamond R \rightarrow (!P \cup R)$. It specifies that during the extent of the starting state and event R , event P must be false. The Existence pattern in pattern scope “Between Q and R ” is described by the formula $(Q \ \&\& \ !R \rightarrow (!R \ W \ (P \ \&\& \ !R)))$, which specifies that during the extent of event Q and event R , event P must become true. The Verifiability is applied in a manner in order to verify properties specified in LTL formulas, we need to define symbols that can be used in formula composition.

The parameters “Revocation”, “Validity” and “Verifiability” have been assigned “Yes” in Badica and Badica (2008) for the sake of presence of formalization of these parameters. Revocation is achieved in a sense that there is a formalization of a property in which a registered participant may submit bids, and then cancel bidding before she can re enter negotiation and start bidding again. The Validity is achieved by LTSA tool that supports a powerful way of checking target system using safety and progress properties defined in the approach. These properties show the verifiability of the approach. Safety property defines that deterministic process asserts that any of the system traces are correct. If there is any error state in Labeled Transition Systems (LTS) with the target system then the safety property is violated.

2.8 Conclusion

Complex activities like Auctions that come across in e-commerce practices frequently are differentiated by a strong interaction of business groups. The study and development of formal modeling is needed for the specification, design, verification and automation of auctions using agents. This provides a strong motivation for the formalization of English auction protocol in multi-agent systems.

For English auction protocol the security aspect is of high importance keeping in view the financial losses caused due to lack of it. There is a strong need to put together and standardize the security parameters involved in this protocol in the form of a model that can provide anonymity, traceability, unforgeability, no framing, unlinkability in different

rounds of an auction, linkability in a round of an auction, fairness, verifiability, one time registration and easy revocation. The evaluation criteria adopted is the analysis of security properties defined in these approaches, due to the reason that the security lapse faced by the established approaches is approximately 87% (Boyd, 2000) collectively.

The formalization of English auction protocol is available only in few languages presenting the abstract picture of this protocol. Still some significant properties had not been taken into account; hence there is a strong need to gather the above mentioned properties and to formalize them.

3

Abstract Model

3. ABSTRACT REPRESENTATION FOR PROPOSED MODEL

Existing formalization approaches do not incorporate security properties like e.g. Lee et al. (2001) have proposed a scheme for EAP that incorporates anonymity, traceability, No Framing, Unlinkability in different rounds of auction, Linkability in a round of auction, Unforgeability, Easy Revocation, Verifiability and One time registration properties, but it is not formalized. The scheme of Lee et al. (2001) has been formalized because it incorporates maximum security properties and hence using Bulletin Board and Signature of Knowledge tools. Lee et al. (2001) recognized a problem in Omote and Miyaji (2001) that the identity of winner cannot be published, so the role of RM cannot be verified because in the winner announcement stage, Registration Manager informs secretly the identity of winner to the vendor. So, the validity of auction is not assured to AM and any bidder. Lee et al. (2001) proposed that in Round setup stage, both the RM and AM execute Randomization operation so that alone RM or AM cannot identify the bidders. It makes the publication of winner's identity possible and a ticket is provided to each bidder so that bidder he can identify his auction ticket, no one else can do that. All the stages of the public auction are verifiable, but there is not any secure channel used in Omote and Miyaji (2001).

3.1 Workflow of the Proposed Methodology

The specifications of EAP have been taken from Lee et al. (2001). Then the construction of Non Deterministic Finite Automata is decided by the behavior of the system. After this, NFA is converted to DFA. Resultant states from Conversion of NFA to DFA include redundancy. These redundant states need to be minimized to get a disjoint set of states. State minimization technique is applied to remove the redundancy among the states. Then this semi formal structure is converted to pure mathematical models, using Z notation. The work flow of the proposed methodology is shown in figure 3.1.

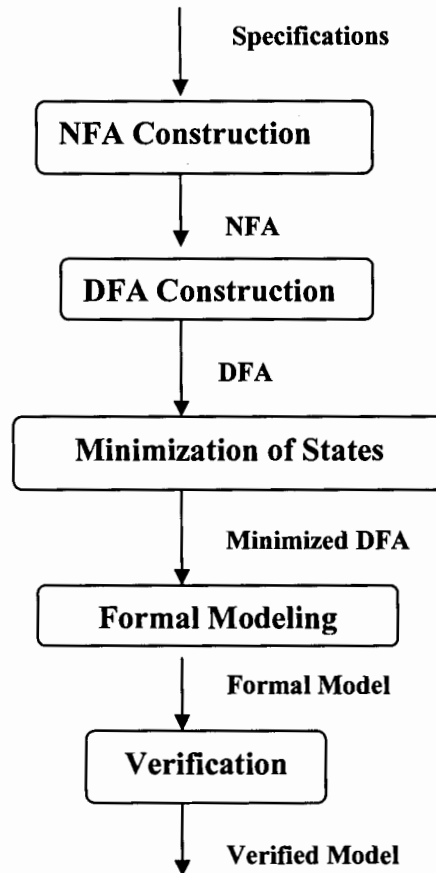


Figure 3.1 Work Flow of Proposed Methodology

3.2 A Finite State Model for the Proposed Scheme

In the proposed scheme, the behavior of the system decides the formation of Non deterministic finite automata. The states of the system have been derived from its entities which are RM, AM and bidder whereas the transitions have been derived from the system flow which describes the information of the processes.

In case of EAP, $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$ represents the set of states, the state invariants are shown in table 3.1, $\Sigma = \{a, b, c, d, f, g, h, \varepsilon\}$, $\delta = \{1x_a \rightarrow 2, 2x_a \rightarrow 2, 2x_b \rightarrow 2, 2x_b \rightarrow 1, 2x_\varepsilon \rightarrow 3, 3x_d \rightarrow 4, 4x_c \rightarrow 3, 4x_d \rightarrow 4, 4x_c \rightarrow 4, 4x_\varepsilon \rightarrow 6, 4x_\varepsilon \rightarrow 9, 5x_\varepsilon$

$\rightarrow 6, 6xf \rightarrow 7, 7xf \rightarrow 7, 7x \varepsilon \rightarrow 9, 8x \varepsilon \rightarrow 9, 9xg \rightarrow 10, 10xg \rightarrow 11, 11x \varepsilon \rightarrow 12, 12xh \rightarrow 13\}$,
 $q_0 = \{1\}, F = \{13\}$

In the perspective of interaction protocols, the agents participate in the conversation use the communicative actions specified by the transitions (Cheremisinov & Cheremisinova, 2006). In this case study agents involved in interaction protocol conversation are RM, AM and Bidder. The graphical representations of negotiations of the agents are defined in a way having fixed structure (Cheremisinov & Cheremisinova, 2006). EAP requirements for formal specifications have been taken from Lee et al. (2001). EAP requirements are then analyzed to identify the accepting states, their respective transitions and the resultant states are shown in table 3.4.

3.2.1 States

The abbreviations that used by the invariants of the Non Deterministic Finite automata of English Auction Protocol are described in the table 3.1.

Table 3.1 Invariants abbreviations of EAP

B	Bidders
RB	Registration Board
RKB	Round Key Board
RKz	Round Keys
BB	Bidding Board
WAB	Winner Announcement Board
WA	Winner Announced

The invariants of the states that have been used in Non Deterministic Finite Automata of English Auction Protocol are described in the table 3.2. These states have been derived to model Non Deterministic Finite Automata of EAP as shown in figure 3.2. Each state have its own description and invariant like in state '1', the invariant is $RB = \emptyset$, means the initial state in which the registration board is empty and bidder has to register for participation in auction. Similarly, state '2' shows that some of the bidders are registered on the registration board. State '3' depicts that round key board is empty, means round keys are

not yet assigned to any bidder to participate in the auction. In the same way as like the mentioned ones, all the states have some invariants as shown in table 3.2.

Table 3.2 State Invariants for NFA of EAP

States	Invariants	Description
1	$RB = \emptyset$	This is the initial state where the Registration Board is empty.
2	$RB \neq \emptyset \wedge RB < n$	This state signifies that a few bidders have been registered.
3	$RKB = \emptyset$	The state where the Round key board is empty.
4	$\#RK > 0 \wedge RKz \in \forall B$	Some of the Round keys have been assigned to bidders in this state.
5	$RKz \in \forall B$	All of the Round keys have been assigned to bidders in this state.
6	$AMB = \emptyset$	The state when Auction Manager Board is empty.
7	$\#Ti > 0 \wedge RNos \in \forall RKz$	When some Round Keys are assigned Random numbers to generate Auction Ticket they are depicted by this state.
8	$RNos \in \forall RKz$	The state when all Round keys have been assigned to Random numbers.
9	$BB = \emptyset$	Empty Bidding Board is shown by this state.
10	$BB \neq \emptyset \wedge BB < n$	In this state the Bidding Board is not full as some bidders haven't yet posted their bids.
11	$BB = n$	This is the state when Bidding Board is full as all bidders have submitted their bids.
12	WAB	All the parameters of bidders will be posted in the winner Announcement board.
13	WA	Final state in which there is a verification of the bidder's parameters and on the basis of which a winner will be announced.

3.2.2 Transitions

The variables that are used for the Operations are described below in the table 3.3. The operation **RegisterBidder** brings change in the RegistrationBoard state. Before this operation the bidder has not yet been registered. This operation brings a bidder to the next state B.

The **RevokeBidderFromRB** operation is triggered by RM. It changes RegistrationBoard and RoundKeyBoard by removing the bidder from the Registration board and Round Key

Board. This operation when executed will not change the state B as there is a state invariant that some bidders might have registered or Registration Board and Round key board are empty.

If a bidder wants to withdraw from the Round Key Board the operation **RevokeBidderfromRKB** is triggered. This operation brings change in RegistrationBoard, RoundKeyBoard, BiddingBoard, AuctionManagerBoard and Bidder. This operation also changed the state from B and E to D.

The operation **GenerateRoundkey** brings a change in Bidder. The bidder's parameters are passed to Auction Manager. The state RoundkeyBoard is evaluated on the basis of components taken from RoundkeyBoard. AuctionManagerBoard is then passed to the Bidding Board as bidders are partially valid bidders now. This operation brings B, D and E states to the state D.

Table 3.3 Operations of EAP

Variables	Operations
a	RegisterBidder
b	RevokeBidderFromRB
c	RevokeBidderFromRKB
d	GenerateRoundKey
f	GenerateAuctionTicket
g	SubmitBids
h	VerifyWinner

The **SubmitBids** operation mentioned in table 3.3 changes the RegistrationBoard, RoundKeyBoard, BiddingBoard, AuctionManagerBoard and Bidder. This operation changes the states G and E to the state H when the Bidding Board is still not full as some bidders haven't yet posted their bids. The same operation changes the state H to state J.

The **VerifyWinner** operation changes the WinnerAnnouncementBoard and BiddingBoard as there is a need to compare the bidder's parameters including Round key, public key and a tj (i.e. combination of 0 and 1 chosen by bidder) along with the values

coming from the Bidding Board to the Winner Announcement Board that are Auction ticket, Random number of a bidder and Round key. This operation changes the state J to the state K which is the final state and winner has been displayed. The operations in table 3.3 have been identified through transitions to model Non Deterministic Finite Automata of EAP as shown in figure 3.2.

3.2.3 Transition table

The transition table for the Non Deterministic Finite Automata of English Auction Protocol is shown in table 3.4.

Table 3.4 State transition table for NFA of EAP

States	a	b	c	d	f	g	h	e
→1(start)	{2}	-	-	-	-	-	-	-
2	{2}	{2,1}	-	-	-	-	-	{3}
3	-	-	-	{4}	-	-	-	-
4	-	-	{3,4}	{4,5}	-	-	-	{6,9}
5	-	-	{4}	-	-	-	-	{6,9}
6	-	-	-	-	{7}	-	-	-
7	-	-	-	-	{7,8}	-	-	{9}
8	-	-	-	-	-	-	-	{9}
9	-	-	-	-	-	{10}	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	{10}	-	{12}
12	-	-	-	-	-	-	{13}	-
13(final)	-	-	-	-	-	-	-	-

Table 3.4 demonstrates the combination of states and transitions. The states are denoted as (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13) in table 3.2 and the operations are depicted as (a, b, c, d, f, g, h, e) in table 3.3. These states have different invariants and pre and post conditions. When operation 'a' is triggered on state '1' in which the registration board is empty, the resultant state is {2} in which registration board is not empty and less than total number of bidders. Similarly, when on state '2' ($RB \neq \emptyset \wedge RB < n$), operation 'a' and 'b' are triggered, the resultant states are {2} and {2, 1} respectively. From state '3' in which round key board is empty, single state {4} in which the some of the round keys

have been assigned to some of the bidders is derived by using the operation 'c'. Similarly, all the states have been derived using the similar pattern.

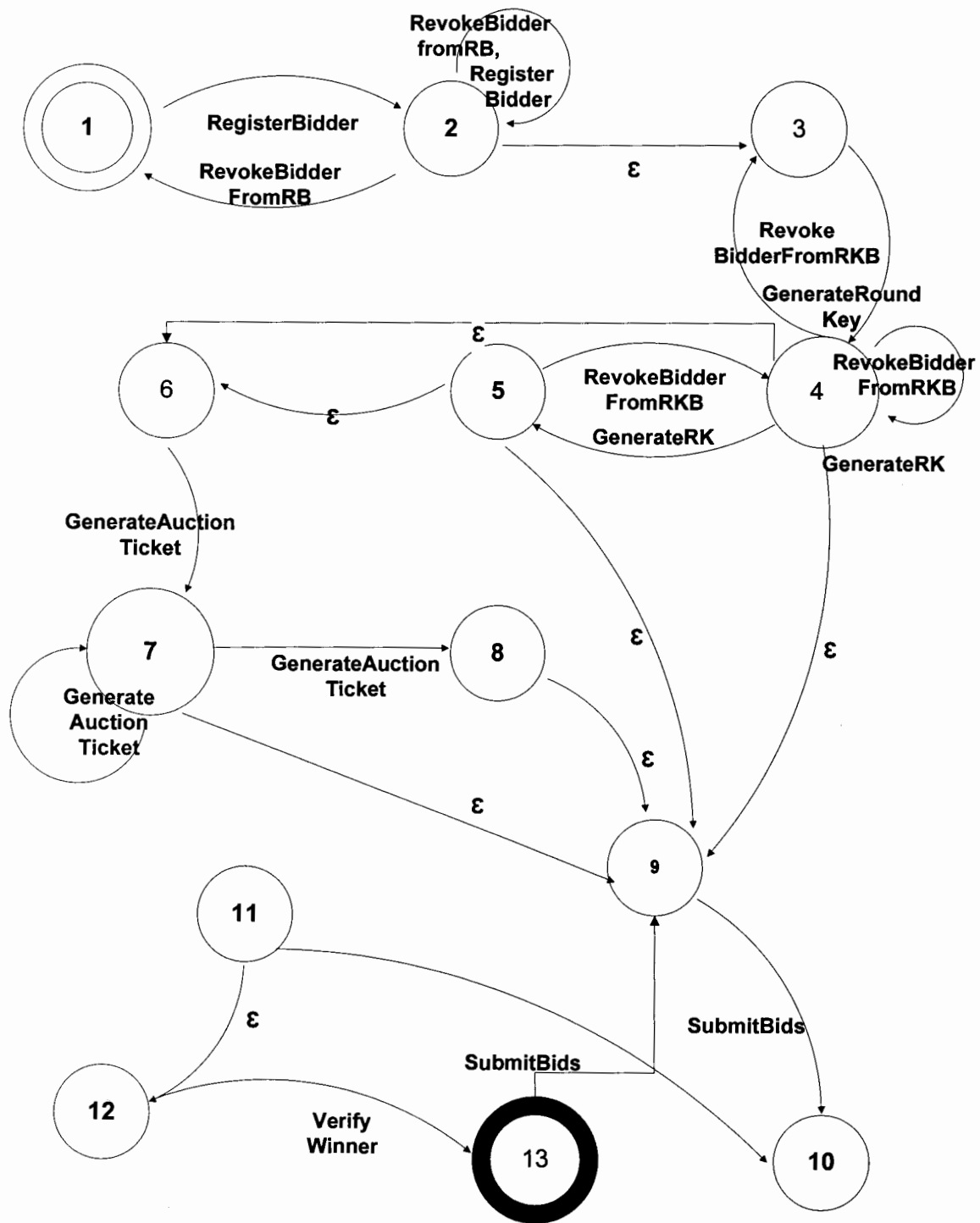


Figure 3.2 NFA of English Auction Protocol

3.3 Making the FSA (Finite State Automata) Deterministic

Non deterministic FSA have been defined with epsilon moves, including back tracking, so it is a slower process. Then, deterministic FSA have been defined for conversation protocol in order to specify the communicative actions among agents in a conversation. Deterministic FSA consists of states and transitions that provides mapping among states.

3.3.1 Conversion of NFA to DFA

The Thompson's subset construction algorithm by Schoop (2005) has been used for the conversion of the Non Deterministic Finite Automata to the Deterministic Finite Automata of an English Auction Protocol.

The ϵ -closure of the initial state '1' is:

$$\epsilon\text{-closure}(1) = \{1\} = A$$

For A:

$$A = \{1\}$$

$$a: \epsilon\text{-closure}(\text{move}(A, a))$$

$$a: \epsilon\text{-closure}(\text{move}(\{1\}, a)) \Rightarrow \epsilon\text{-closure}(\{2\}) = \{2, 3\} = B$$

$$b: \epsilon\text{-closure}(\text{move}(\{1\}, b)) \Rightarrow \epsilon\text{-closure}(\{\}) = \{\} = \emptyset$$

$$c: \epsilon\text{-closure}(\text{move}(\{1\}, c)) \Rightarrow \epsilon\text{-closure}(\{\}) = \{\} = \emptyset$$

$$d: \epsilon\text{-closure}(\text{move}(\{1\}, d)) \Rightarrow \epsilon\text{-closure}(\{\}) = \{\} = \emptyset$$

$$f: \epsilon\text{-closure}(\text{move}(\{1\}, f)) \Rightarrow \epsilon\text{-closure}(\{\}) = \{\} = \emptyset$$

$$g: \epsilon\text{-closure}(\text{move}(\{1\}, g)) \Rightarrow \epsilon\text{-closure}(\{\}) = \{\} = \emptyset$$

$$h: \epsilon\text{-closure}(\text{move}(\{1\}, h)) \Rightarrow \epsilon\text{-closure}(\{\}) = \{\} = \emptyset$$

After checking the ϵ -closure with all the operations on the first state, we get another state B.

For B:

$$B = \{2, 3\}$$

a: ϵ -closure (move (B, a))

a: ϵ -closure (move ({2, 3}, a)) $\Rightarrow \epsilon$ -closure ({2}) = {2, 3} = B

b: ϵ -closure (move ({2, 3}, b)) $\Rightarrow \epsilon$ -closure ({2, 1}) = {2, 3, 1} = C

c: ϵ -closure (move ({2, 3}, c)) $\Rightarrow \epsilon$ -closure ({4}) = {4, 6, 9} = D

d: ϵ -closure (move ({2, 3}, d)) $\Rightarrow \epsilon$ -closure ({4}) = {4, 6, 9} = D

f: ϵ -closure (move ({2, 3}, f)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

g: ϵ -closure (move ({2, 3}, g)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

h: ϵ -closure (move ({2, 3}, h)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

There are number of states determined after this computation. The states are B, C and D.

For C:

C= {2, 3, 1}

a: ϵ -closure (move(C, a))

a: ϵ -closure (move ({2, 3, 1}, a)) $\Rightarrow \epsilon$ -closure ({2}) = {2, 3} = B

b: ϵ -closure (move ({2, 3, 1}, b)) $\Rightarrow \epsilon$ -closure ({2, 1}) = {2, 3, 1} = C

c: ϵ -closure (move ({2, 3, 1}, c)) $\Rightarrow \epsilon$ -closure ({4}) = {4, 6, 9} = D

d: ϵ -closure (move ({2, 3, 1}, d)) $\Rightarrow \epsilon$ -closure ({4}) = {4, 6, 9} = D

f: ϵ -closure (move ({2, 3, 1}, f)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

g: ϵ -closure (move ({2, 3, 1}, g)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

h: ϵ -closure (move ({2, 3, 1}, h)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

Now the states B, C and D are identified.

For D:

D= {4, 6, 9}

a: ϵ -closure (move (D, a))

a: ϵ -closure (move ({4, 6, 9}, a)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

b: ϵ -closure (move ({4, 6, 9}, b)) $\Rightarrow \epsilon$ -closure ({}) = {} = \emptyset

c: ϵ -closure (move ({4, 6, 9}, c)) $\Rightarrow \epsilon$ -closure ({4, 3}) = {3, 4, 6, 9} = E

d: ϵ -closure (move ({4, 6, 9}, d)) $\Rightarrow \epsilon$ -closure ({4, 5}) = {4, 6, 9, 5} = F

f: ϵ -closure (move ($\{4, 6, 9\}$, f)) \Rightarrow ϵ -closure ($\{7\}$) = $\{7, 9\}$ = G

g: ϵ -closure (move ($\{4, 6, 9\}$, g)) \Rightarrow ϵ -closure ($\{10\}$) = $\{10\}$ = H

h: ϵ -closure (move ($\{4, 6, 9\}$, h)) = ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

The two states E and F are discovered after doing ϵ -closure on the D state.

For E:

E = $\{4, 6, 9, 3\}$

a: ϵ -closure (move (E, a))

a: ϵ -closure (move ($\{4, 6, 9, 3\}$, a)) \Rightarrow ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

b: ϵ -closure (move ($\{4, 6, 9, 3\}$, b)) \Rightarrow ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

c: ϵ -closure (move ($\{4, 6, 9, 3\}$, c)) \Rightarrow ϵ -closure ($\{4\}$) = $\{4, 6, 9\}$ = D

d: ϵ -closure (move ($\{4, 6, 9, 3\}$, d)) \Rightarrow ϵ -closure ($\{4, 5\}$) = $\{4, 6, 9, 5\}$ = F

f: ϵ -closure (move ($\{4, 6, 9, 3\}$, f)) \Rightarrow ϵ -closure ($\{7\}$) = $\{7, 9\}$ = G

g: ϵ -closure (move ($\{4, 6, 9, 3\}$, g)) \Rightarrow ϵ -closure ($\{10\}$) = $\{10\}$ = H

h: ϵ -closure (move ($\{4, 6, 9, 3\}$, h)) \Rightarrow ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

By performing ϵ -closure on the state E, the new states are D and F.

For F:

D = $\{4, 5, 6, 9\}$

a: ϵ -closure (move (F, a))

a: ϵ -closure (move ($\{4, 5, 6, 9\}$, a)) \Rightarrow ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

b: ϵ -closure (move ($\{4, 5, 6, 9\}$, b)) \Rightarrow ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

c: ϵ -closure (move ($\{4, 5, 6, 9\}$, c)) \Rightarrow ϵ -closure ($\{4, 3\}$) = $\{3, 4, 6, 9\}$ = E

d: ϵ -closure (move ($\{4, 5, 6, 9\}$, d)) \Rightarrow ϵ -closure ($\{4, 5\}$) = $\{4, 6, 9, 5\}$ = F

f: ϵ -closure (move ($\{4, 5, 6, 9\}$, f)) \Rightarrow ϵ -closure ($\{7\}$) = $\{7, 9\}$ = G

g: ϵ -closure (move ($\{4, 5, 6, 9\}$, g)) \Rightarrow ϵ -closure ($\{10\}$) = $\{10\}$ = H

h: ϵ -closure (move ($\{4, 5, 6, 9\}$, h)) \Rightarrow ϵ -closure ($\{\}$) = $\{\}$ = \emptyset

The derived states by doing ϵ -closure on F state are E, F, G and H states.

For G:

$$G = \{7, 9\}$$

a: ϵ -closure (move (E, a))

$$a: \epsilon\text{-closure (move (\{7, 9\}, a))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$b: \epsilon\text{-closure (move (\{7, 9\}, b))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$c: \epsilon\text{-closure (move (\{7, 9\}, c))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$d: \epsilon\text{-closure (move (\{7, 9\}, d))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$f: \epsilon\text{-closure (move (\{7, 9\}, f))} \Rightarrow \epsilon\text{-closure (\{7, 8\})} = \{7, 9, 8\} = I$$

$$g: \epsilon\text{-closure (move (\{7, 9\}, g))} \Rightarrow \epsilon\text{-closure (\{10\})} = \{10\} = H$$

$$h: \epsilon\text{-closure (move (\{7, 9\}, h))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

For getting the new states ϵ -closure is performed and new states are I and H.

For H:

$$H = \{10\}$$

a: ϵ -closure (move (H, a))

$$a: \epsilon\text{-closure (move (10, a))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$b: \epsilon\text{-closure (move (10, b))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$c: \epsilon\text{-closure (move (10, c))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$d: \epsilon\text{-closure (move (10, d))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$f: \epsilon\text{-closure (move (10, f))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

$$g: \epsilon\text{-closure (move (10, g))} \Rightarrow \epsilon\text{-closure (\{11\})} = \{11, 12\} = J$$

$$h: \epsilon\text{-closure (move (10, h))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

The new identified state is J.

For I:

$$I = \{7, 9, 8\}$$

a: ϵ -closure (move (I, a))

$$a: \epsilon\text{-closure (move (\{7, 9, 8\}, a))} \Rightarrow \epsilon\text{-closure (\{\})} = \{\} = \emptyset$$

b: ϵ -closure (move ($\{7, 9, 8\}$, b)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

c: ϵ -closure (move ($\{7, 9, 8\}$, c)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

d: ϵ -closure (move ($\{7, 9, 8\}$, d)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

f: ϵ -closure (move ($\{7, 9, 8\}$, f)) $\Rightarrow \epsilon$ -closure ($\{7, 8\}$) = $\{7, 9, 8\}$ = I

g: ϵ -closure (move ($\{7, 9, 8\}$, g)) $\Rightarrow \epsilon$ -closure ($\{10\}$) = $\{10\}$ = H

h: ϵ -closure (move ($\{7, 9, 8\}$, h)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

The derived states are I and H.

For J:

J= $\{11, 12\}$

a: ϵ -closure (move (J, a))

a: ϵ -closure (move ($\{11, 12\}$, a)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

b: ϵ -closure (move ($\{11, 12\}$, b)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

c: ϵ -closure (move ($\{11, 12\}$, c)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

d: ϵ -closure (move ($\{11, 12\}$, d)) $\Rightarrow \epsilon$ -closure ($\{\}$) = $\{\}$ = \emptyset

f: ϵ -closure (move ($\{11, 12\}$, f)) $\Rightarrow \epsilon$ -closure ($\{7, 8\}$) = $\{7, 9, 8\}$ = I

g: ϵ -closure (move ($\{11, 12\}$, g)) $\Rightarrow \epsilon$ -closure ($\{10\}$) = $\{10\}$ = H

h: ϵ -closure (move ($\{11, 12\}$, h)) $\Rightarrow \epsilon$ -closure ($\{13\}$) = $\{13\}$ = K

The numbers of states after performing the ϵ -closure on J state are I, H and K states.

K= $\{13\}$ =Final

NFA is used to check an accepting combination of characters, it involves backtracking to check for choices that it did not make earlier. For the sake of uninterrupted communication among all accepting states, certain states are connected with other states using epsilon moves. Although, these empty transitions makes NFA slower, where DFA has a unique path for each accepting string and backtracking is not involved. This makes them faster than NFA. This conversion of NFA to DFA is performed using Thompson's subset construction algorithm [47]. Firstly, ϵ -closure of initial state of NFA is computed, and then all the states reachable from initial state with epsilon moves are identified to get

the new state of DFA. Then all the operations defined in table 3.3 are applied on each state to get the new state of DFA. Similarly, all the DFA states are calculated by applying Thompson's algorithm on all NFA states.

NFA is converted to DFA to fill the epsilon moves depicted by ϵ in Σ of NFA. Where $Q = \{A, B, D, E, G, H, J, K, \emptyset\}$, $\delta = \{A \times a \rightarrow B, A \times \{b, c, d, f, g, h\} \rightarrow \emptyset, B \times \{a, b\} \rightarrow B, B \times \{f, g, h\} \rightarrow \emptyset, B \times \{c, d\} \rightarrow D, D \times d \rightarrow D, D \times c \rightarrow E, D \times f \rightarrow G, D \times \{a, b, h\} \rightarrow \emptyset, E \times \{c, d\} \rightarrow D, E \times f \rightarrow G, E \times g \rightarrow H, H \times g \rightarrow J, H \times \{a, b, c, d, f\} \rightarrow \emptyset, G \times f \rightarrow G, G \times g \rightarrow H, G \times \{a, b, c, d, h\} \rightarrow \emptyset, J \times \{a, b, c, d, f, g, h\} \rightarrow \emptyset, K \times \{a, b, c, d, h\} \rightarrow \emptyset\}$, $q_0 = \{A\}$, $F = \{K\}$.

For all three agents of English Auction protocol namely Auction Manager, Registration Manager and Bidder, system state is represented by five components. The unique entities in English Auction Protocol were Bidder Registration Board, Round Key Board, Auction Ticket Board, Winner Announcement Board, and Bidding Board. Certain operations shown in table 3.3 are performed on the above mentioned states for moving to the new/next states.

Bidder Registration Board: This state has been identified, when a bidder registers as a participant for auction. Bidder will have his own private and public keys. The agent Registration Manager will perform this by using the 'Register bidder (a)' operation. When the bidder wants to quit the auction just after the registration, he uses the 'revoke bidder from Registration Board (f)' to quit from the auction.

Round Key Board: Agent Registration Manager works out the Bidder agent's parameters and generates the round key by using 'Generate round key (d)' bidder agent from the registration manager board. The operation 'generate auction ticket (f)' will be used by the Auction Manager.

Auction Ticket Board: The auction ticket is computed by Auction manager using round key of bidder agent from the registration manager board. The operation 'generate auction ticket (f)' have been used by the Auction Manager.

Bidding Board: When Bidder agent wants to participate in the auction he posts the bidding information on the Bidding Board using the operation 'submit bids (g)'.

Winner Announcement Board: At the end of every auction a winner is announced. This winner is selected after the verification process of the valid bidder agents.

3.3.2 Derived States

A: {1}

B: {2, 3}

C: {2, 3, 1}

D: {4, 6, 9}

E: {4, 6, 9, 3}

F: {4, 5, 6, 9}

G: {7, 9}

H: {10}

I: {7, 9, 8}

J: {11, 12}

K: {13} Final state

∅: Empty State

These states have been derived to model DFA of English Auction Protocol, but there are some states which are redundant. For removing this duplication and redundancy, states minimization technique is applied on the states having similar destinations in DFA of EAP to get minimized and disjoint set of states. Finally minimized Deterministic Finite Automata of English Auction Protocol as shown in figure 3.3.

3.3.3 Transition Table for DFA of English Auction Protocol

The transition table for the Deterministic Finite Automata of English Auction Protocol is shown in table 3.5.

Table 3.5 State transition table for DFA of EAP

D States	a	b	c	d	f	g	h
-A	B	∅	∅	∅	∅	∅	∅
B	B	C	D	D	∅	∅	∅
C	B	C	D	D	∅	∅	∅
D	∅	∅	E	F	G	H	∅
E	∅	∅	D	F	G	H	∅
F	∅	∅	E	F	G	H	∅
G	∅	∅	∅	∅	I	H	∅
H	∅	∅	∅	∅	∅	J	∅
I	∅	∅	∅	∅	I	H	∅
J	∅	∅	∅	∅	∅	∅	K
+K	∅	∅	∅	∅	∅	∅	∅

3.4 States Minimization

When there are number of similar states determined after performing operations on different states, then they can be combined as a single state and this can be done by using the states minimization technique. Resultant states from Conversion of NFA to DFA include redundancy. These redundant states need to be minimized in to disjoint and non overlapping set of states of above mentioned protocol. State minimization technique is applied to remove the redundancy among the states. The identified redundant states determined in DFA of English Auction Protocol (EAP) like $B \cup C \Rightarrow B$, $D \cup F \Rightarrow D$, and $G \cup I \Rightarrow G$. B, D and G are the final set of disjoint and minimized states for EAP.

i) $B \cup C \Rightarrow BC$

ii) $D \cup F \Rightarrow DF$

iii) $G \cup I \Rightarrow GI$

The transition table for the Deterministic Finite Automata of an English Auction Protocol after applying the states minimization technique is shown in table 3.6.

Table 3.6 State transition table for Minimized DFA

D States	a	b	c	d	f	g	h
-A	BC	∅	∅	∅	∅	∅	∅
BC	BC	BC	DF	DF	∅	∅	∅
DF	∅	∅	E	DF	GI	H	∅
E	∅	∅	DF	DF	GI	H	∅
GI	∅	∅	∅	∅	GI	H	∅
H	∅	∅	∅	∅	∅	J	∅
J	∅	∅	∅	∅	∅	∅	K
Null	∅	∅	∅	∅	∅	∅	∅
+K	∅	∅	∅	∅	∅	∅	∅

The table 3.6 illustrates that by applying the operations a, b, c, d, f, g and h on the state A of DFA of EAP, only the resultant state BC is prompted, similarly by applying the operations a, b, c and d on BC of DFA of EAP, the resultant states BC and DF are derived. The resultant states E, DF, GI and H are triggered when operations c, d, f and g are applied DF state. By executing the operations a, b, c, d, f, g, and h on GI state, the resultant states are GI and H. The state H along with the operation g has derived the state J. By applying all the operations on the Null state, the empty state (Null state) is triggered. Finally, all the operations performed on the final state K, the resultant state is also Null.

The Deterministic Finite Automata for English Auction Protocol is shown in figure 3.3.

As result of States minimization, some states are mapped as follows:

- BC=B
- DF=D
- GI=G

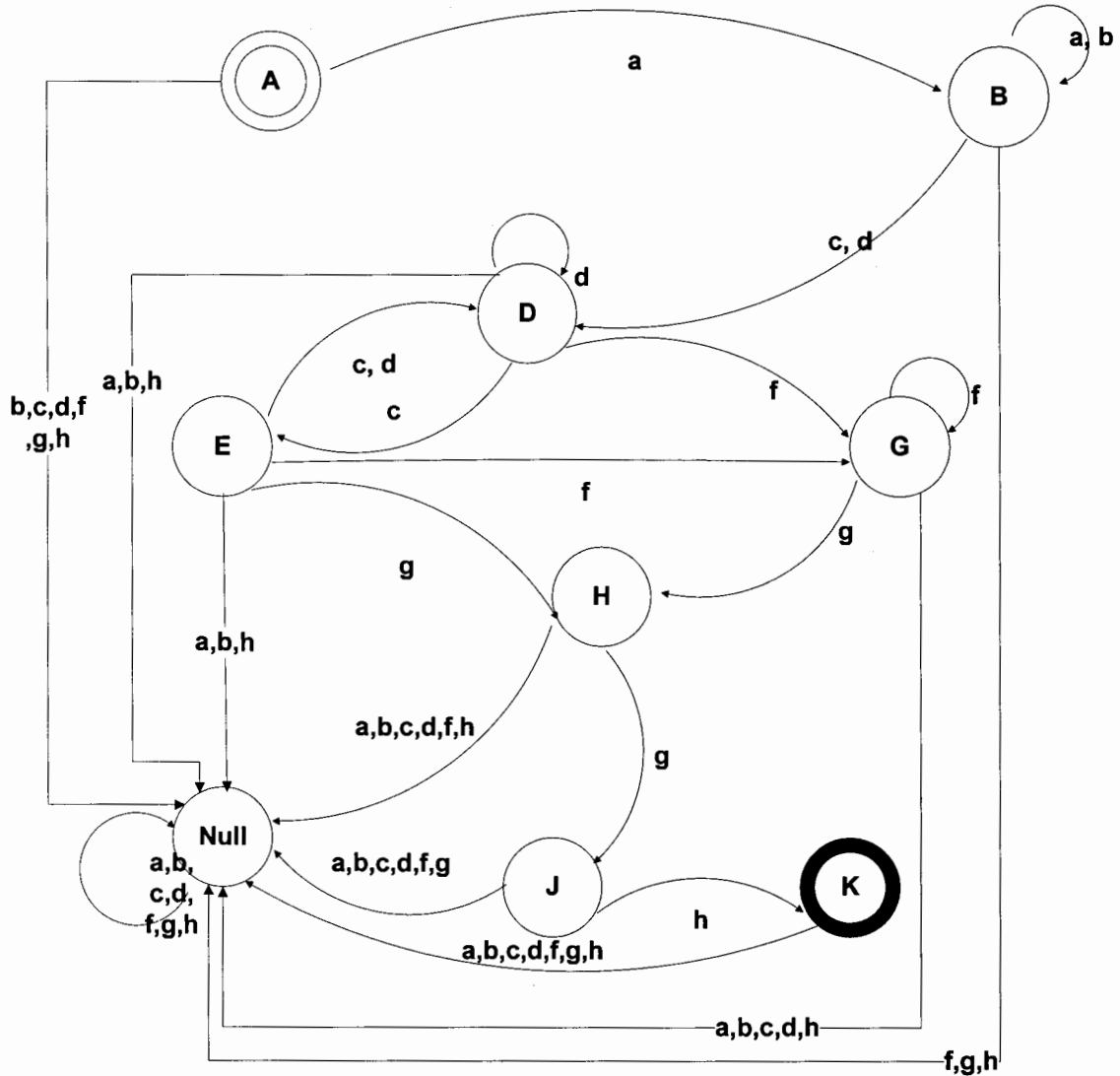


Figure 3.3 Minimized Deterministic Finite Automata of EAP

3.4.1 State Invariants

The invariants that are used for the states of a Deterministic Finite Automata of English Auction Protocol described in the table 3.7.

Table 3.7 State Invariants for DFA of EAP

Variables	States	Explanation
-A	$RB = \emptyset$	Initial state in which Registration Board is empty.
B	$RB \neq \emptyset \wedge RB < n \cup$ $RKB = \emptyset \cup RB = \emptyset$	Some bidders might have registered or Registration Board is empty. Round key board is empty.
D	$\#RK > 0 \wedge RKz \in$ $\forall B \cup AMB = \emptyset$ $\cup RKz \in \forall B \cup$ $BB = \emptyset$	All or some of Round keys have been assigned to bidders, Auction Manager Board and Bidding Board is empty.
E	$RKB = \emptyset$ $\cup \#RK > 0 \wedge RKz \in$ $\forall B \cup AMB = \emptyset$	Only some of Round keys have been assigned to bidders, Auction Manager Board and Round Key Board are empty.
G	$\#Ti > 0 \wedge RNos \in \forall$ $RKz \cup RNos \in$ $\forall RKz \cup BB = \emptyset$	All or some of Round Keys are being assigned a Random number to generate Auction Ticket, Bidding Board is empty.
H	$BB \neq \emptyset \wedge BB < n$	Bidding Board is not full as some bidders haven't posted their bids yet.
J	$BB = n \cup WAB$	Bidding Board is full as all bidders have submitted their bids.
+K	WA	Final state in which there is a verification of the bidder's parameters and then the winner will be announced.

'A' is the initial state from which the bidding starts, hence the registration board is empty at this time, and bidders have to register to participate in the auction.

The next state 'B' has been derived from the state 'A', in which some bidders might have registered for participation in auction or Registration Board is still empty as no bidder has registered as yet. The Round key board is also empty, as when all the bidders will be registered then the round keys will be generated for the bidders.

The new state 'D' is formed from state 'B' and in state 'D', all or some of Round keys have been assigned to bidders, at the same time Auction Manager Board and Bidding Board are empty, as auction tickets have not been generated as yet and bidders can't submit their bids before getting their auction tickets and round keys.

The next state is 'E' where only some of Round keys have been assigned to bidders, Auction Manager Board and Round Key Board are empty.

The state 'H' has a constraint that Bidding Board is not full as some bidders haven't posted their bids yet, will be generated when the operation 'g' will trigger on 'G' and 'E' states.

The new state 'J' is attained from state 'H' in which the Bidding Board is full as all bidders have submitted their bids.

The state 'J' is converted to state 'K' which is the final state here verification of the bidder's parameters is done and then a winner will be announced.

4

Formal Model



4. FORMAL MODEL FOR PROPOSED SYSTEM

This chapter illustrates the formal model of the proposed system. It presents the formal framework for English Auction Protocol using Z notation. Here we convert prescribed semi formal structures to pure mathematical models, using Z notation. The syntax provided by Z enables us to write conceptual level specifications which are close to the user requirement but have no relevance at the implementation stage [1]. In this chapter, we first give global declarations and definitions and then present system static and dynamic model. The formal specifications of the English Auction Protocol are:

4.1 Global Declarations and Definitions

The global declarations, abbreviation definitions and axiomatic descriptions used in the formal specifications of English Auction Protocol are described below.

4.1.1 CHAR

[*CHAR*] is a generic type. It represents the set of all the characters that are used to post the identities of the bidders.

4.1.2 TEXT

TEXT == seq *CHAR*

TEXT: The sequence of all the characters used to write the name of the bidders.

4.1.3 Integers

| *p, q, g, n*: \mathbb{Z}

This axiomatic definition has no predicate and these integers are used to compute the public key and to encrypt it. These are used in the Registration Manager's bulletin board.

4.1.4 Power

$$\begin{array}{|l}
 \text{power: } \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z} \\
 \hline
 \forall a, b: \mathbb{Z} \\
 \cdot (b \leq 1 \Rightarrow \text{power}(a, b) = a) \\
 \wedge (b > 1 \Rightarrow \text{power}(a, b) = a * \text{power}(a, (b - 1)))
 \end{array}$$

Power function is a recursive definition calculating the exponent by taking two integers as inputs and provides one integer as an output. It is used by RoundKeyBoard schema and AuctionManagerBoard schema, when Auction manager and Registration manager computes the Auction Ticket and Round Key for the bidder.

4.1.5 Compute Public Key

$$\begin{array}{|l}
 \text{ComputePublicKey: } \mathbb{Z} \rightarrow \mathbb{Z} \\
 \hline
 \neg p = 0 \\
 \forall x: \mathbb{Z} \cdot \text{ComputePublicKey } x = \text{power}(g, x) \bmod p
 \end{array}$$

ComputePublicKey calculates public key for the bidders, It takes a single integer as an input and the public key of type integer is an output which has been calculated by using the power function and it is used in AuctionManagerBoard schema for the calculation of the private key of the Auction Manager.

4.1.6 Encrypt

$$\begin{array}{|l}
 \text{Encrypt: } \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z} \\
 \hline
 \neg p = 0 \\
 \forall m, x: \mathbb{Z} \cdot \text{Encrypt}(m, x) = \text{power}(m, x) \bmod p
 \end{array}$$

The Encrypt function takes two integers as inputs (m and x) and encrypts them by using the power function which is the output and it is used in the RegistrationBoard schema in order to compute the signature of the bidder for security purpose.

4.1.7 Maximum

$$\text{maximum}: \mathcal{P} \text{ BiddingBoard} \rightarrow \text{BiddingBoard}$$

$$\forall B: \mathcal{P} \text{ BiddingBoard}$$

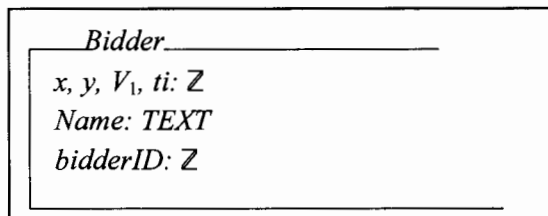
$$\bullet \exists b: B \bullet (\forall b_1: B \bullet b \text{ . bidprice} \geq b_1 \text{ . bidprice}) \Rightarrow \text{maximum } B = b$$

The maximum function has been designed to get the maximum bid price from all the bids using BiddingBoard; Input for this function is the power set of bids. The output will be the maximum bid among all those bids and is used in Winner Announcement Board schema and Verify Bidder schema as when the maximum bid will be identified.

4.2 Static Formal Model

The static model consists of all state schemas of the English Auction system.

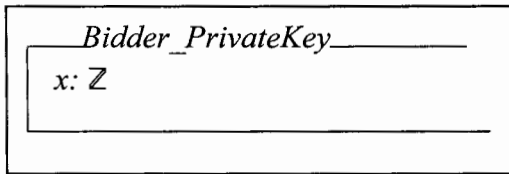
4.2.1 Bidder



Schema Name: Bidder

Description: Bidder schema consists of five components, i.e., ‘y’ denoting unique identifiers which will act as public key, and it is needed in GenerateRoundKey and RevokeBidderfromRKB schemas. ‘V₁’ denotes the signature to indicate validity of the bids and ‘Name’ as the name of the bidder which will be needed further in schemas like RegistrationBoard and BiddingBoard schema. This schema describes the unique Bidder agent that keeps the information of the bidder in order to participate in the auction process.

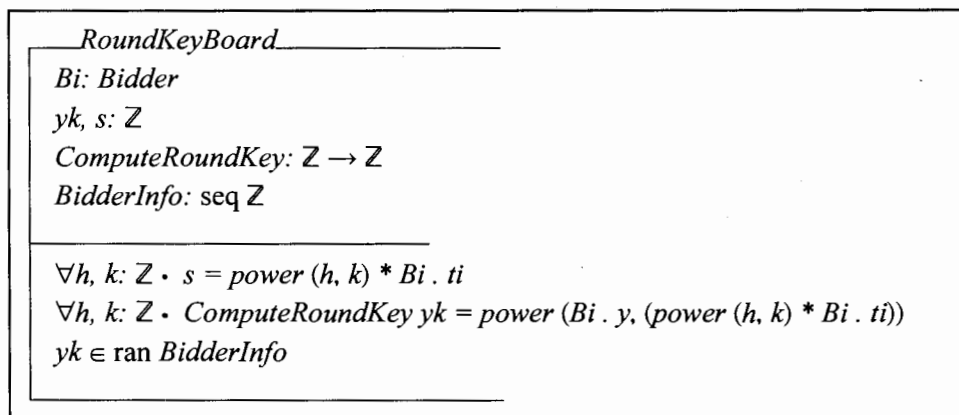
4.2.2 Bidder's Private Key



Schema Name: Bidder_PrivateKey

Description: Private Key describes the unique bidder agent and has a private key 'x', only known to the bidder for the sake of anonymity.

4.2.3 Round Key Board



Schema Name: RoundKeyBoard

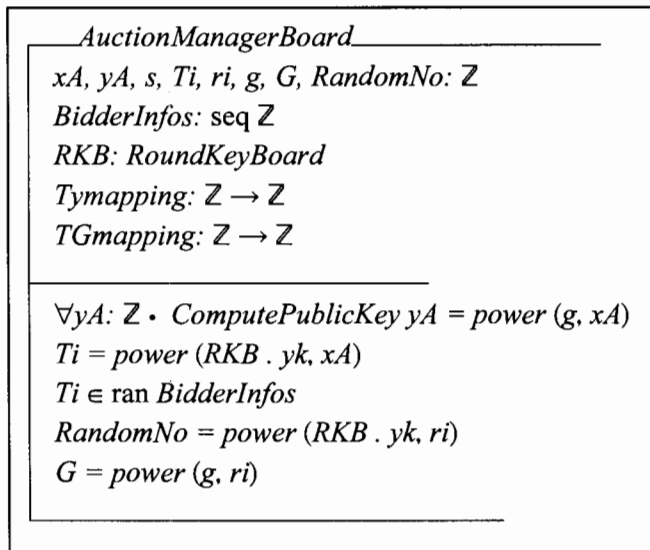
Description: A RoundKeyBoard is written by the Registration Manager and Registration Manager computes the Round keys for the registered bidders using ComputeRoundKey function. This schema depicts the state of round keys. It tells how many round keys were generated by that includes the operation schemas that can change the state of the system by changing the values of the variables in state schemas and then these become Registration Manager (RM) for the number of registered bidders, as the bidders can't be able to participate in the auction without-having Round keys. This information is stored in a sequence 'BidderInfo', so that the bidderID can be matched with this sequence to check the validity of the bidder. The integer's' is calculated here

using Bidder's value 'ti'. This's' will be used in DisplayWinnerAnnouncementBoard schema.

Invariants:

- The variable's' should be calculated by using the Bidder's value 'ti'.
- The elements of ComputeRoundKey 'yk' should be calculated using bidder's public key 'y' and the element of bidder 'ti'.
- The elements of 'yk' set should be the same as elements stored in database in BidderInfo sequence.

4.2.4 Auction Manager Board



Schema Name: AuctionManagerBoard

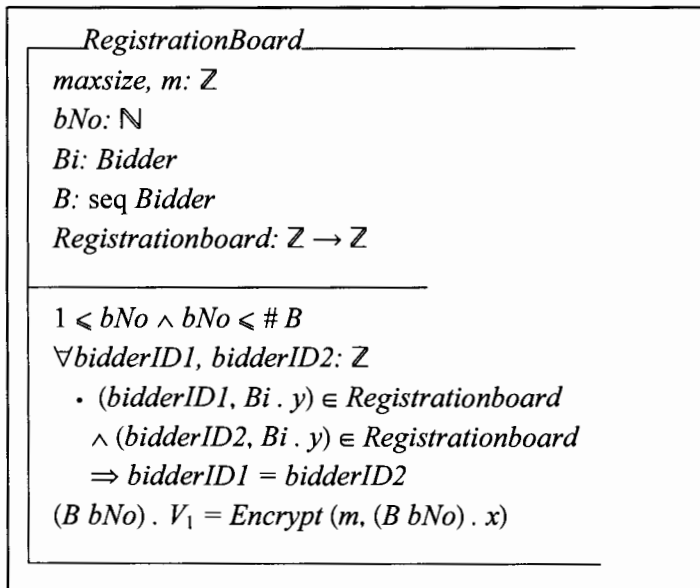
Description: Auction Manager Board schema portrays the auction tickets for every valid bidder listed in the round key board of Registration Manager (RM). This is performed by the Auction Manager (AM). AM can update the bulletin board and computes the auction tickets 'Ti' for every valid bidder against the round key 'yk' listed in the Round Key Board. After that, Ti (ticket identifier) is stored in the sequence 'BidderInfos'. Ticket identifier by Auction Manager is calculated by Diffie-Helman key agreement, which is the shared key between the bidder and the Auction Manager. 'RandomNo' is an integer provided by the AM to each valid bidder. 'Tymapping' and 'TGmapping' functions are

declared here to show the correspondence among the RandomNo, G and Ti of the particular bidder for the ease of verification.

Invariants:

- The elements of ComputePublicKey 'yA' set should be the same as elements calculated by computing the Auction manager's parameters g and private value 'xA' using power function.
- The set Ti, the Auction Tickets can be calculated using the Roundkey of the bidder and the secret value of Auction Manager xA.
- The elements of 'Ti' set should be the same as elements stored in database in BidderInfos sequence.
- The RandomNo for each bidder should be calculated using the Round key given by Registration Manager and the random no 'ri' by the Auction Manager within power function.
- The element G can be computed by using the elements 'g' and 'ri' by the Auction Manager.

4.2.5 Registration Board



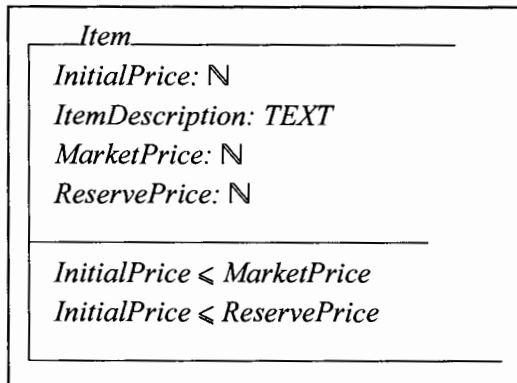
Schema Name: RegistrationBoard

Description: Registration Board schema explains the working of RM and illustrates the identities and public keys of the registered bidders. 'Registration board' function takes 't' as an input and returns the integer for verification of each 'bidderID' for which there must be a unique public key 'y' and the signature of the bidder, encrypted in 'V₁'. The integers 'maxsize' and 'm' are used in this schema for that signature 'V₁'.

Invariants:

- The number of bidders 'bNo' must be between 1 and the cardinality of total number of Bidders.
- If the two bidders that have different ID along with the same public key 'y' belongs to the elements of set Registrationboard, then that bidderID1 must be the same bidderID2 as there must not be the same public key for two bidders.
- The private signature of each bidder can be calculated by using the private key 'x' of the bidder and the bid value 'm' using Encrypt function.

4.2.6 Item



Schema Name: Item

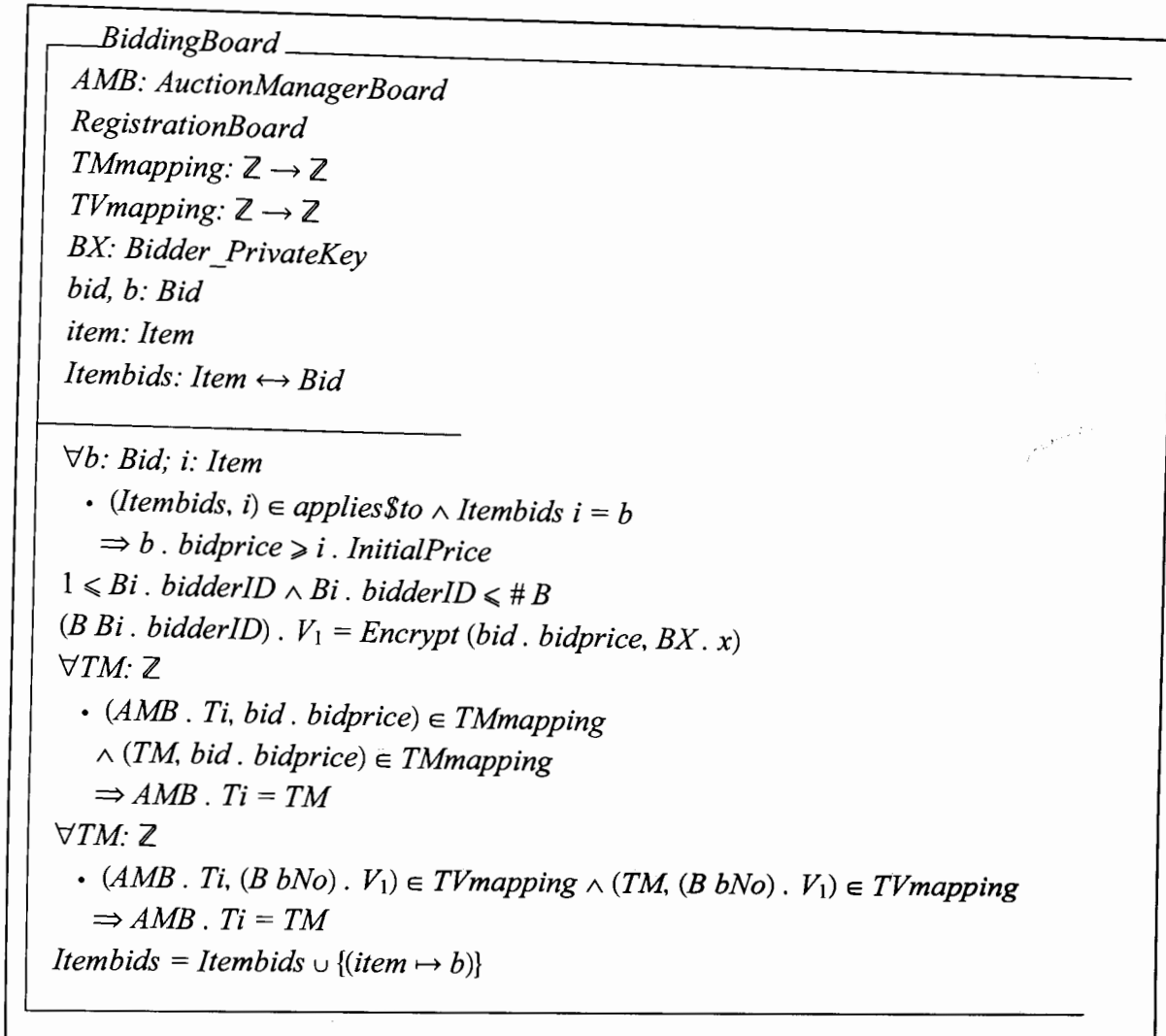
Description: Item consists of four components, hence, shows each item to be auctioned that has an 'InitialPrice' at which the auction starts, the description of the item through component 'ItemDescription', its market price through component 'MarketPrice' and the 'Reserve Price' component set by the Auction Manager.

Invariants:

- Initial price of the item should be less than equal to the market price of that item.

- Initial price should also be less than or equal to the reserve price of that item.

4.2.7 Bidding Board



Schema Name: BiddingBoard

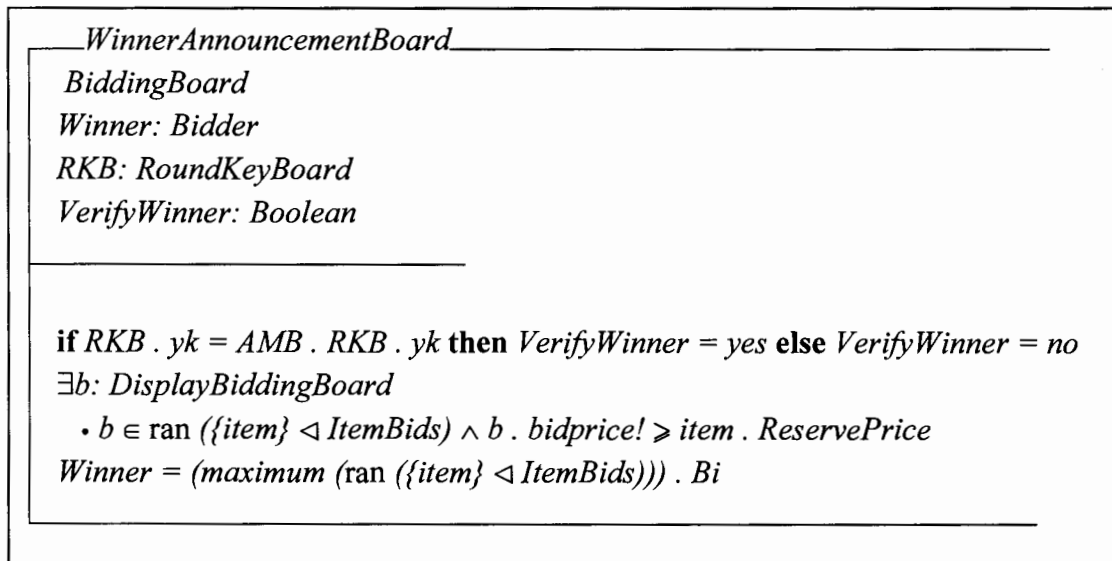
Description: Bidding Board schema shows the bidder's posts about bidding information. Only those bids are posted that are higher than the previous ones and the posting of bids cannot be prevented by anybody. The two Boolean components 'IdentifyTicket' and 'CheckRoundKey' verify that 'bidderID' belongs to legitimate Round Keys and Auction Tickets respectively. Bidders are required to identify their Round Keys and Auction

Tickets in this Bidding Board. ‘TMmapping’ and ‘TVmapping’ are designed to prepare bidder’s bid information i.e T_i , m and V_1 to be posted on the Bidding Board.

Invariants:

- If there is an Auction Ticket against the bidderID then the ticket will be identified and will be assigned ‘yes’.
- If there is a Round key against the bidderID then this means that Round key has been assigned to the bidder therefore will be marked ‘yes’.
- If there are three elements Auction Ticket ‘ T_i ’, bid value ‘ m ’ and the signature of the bidder ‘ V_1 ’ present against each bidder, this means that there is a correspondence of the values of the bidder, for that purpose the above two mappings TMmapping and TVmapping are used.

4.2.8 Winner Announcement Board



Schema Name: WinnerAnnouncementBoard

Description: The WinnerAnnouncementBoard schema expresses winner dependent secret random number and information by AM and RM, respectively using the components. Item is of type Item and Winner is the bidder among Number of Bidders. ‘VerifyWinner’ is of type Boolean, while ‘ItemBids’ is a function of which item is the

input and the output is evaluated from the BiddingBoard schema. The auction ends up successfully if:

- There is a maximum bid among the set of all valid bids and;
- It is greater than the reserve price

The maximum bid is checked for:

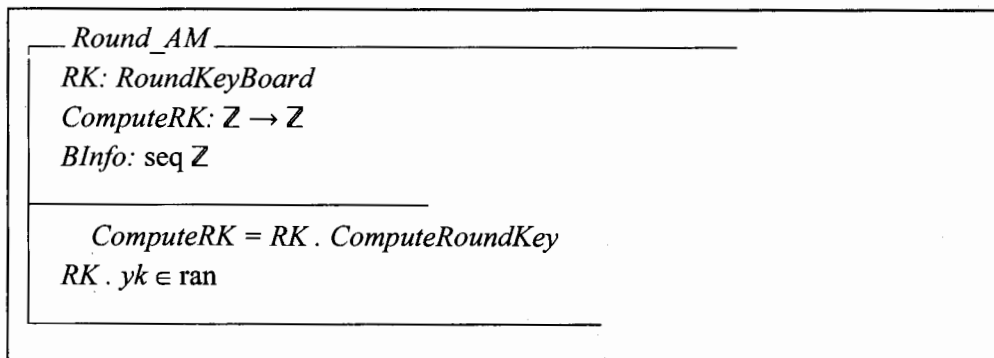
- It belongs to the range of items already defined
- All the number of bids against that item
- Its bid price which should be greater than the reserve price

Finally, the maximum bid from this scenario will be the winning bid.

Invariants:

- If for each item 'i' there is a bid 'b' then the bid price must be greater than the initial price of that item.
- The Round key that is computed on the Round key board will be further processed in the Auction Manager Board. If it is same then we can verify the winner, otherwise winner can't be verified.
- The bidder's bid 'b' must belong to the number of bids against each item and that bid price must be greater than item's Reserve price set by the Auction Manager.
- The winning bidder will be decided on the basis of highest bid.

4.2.9 Access of Round Keys for Auction Manager



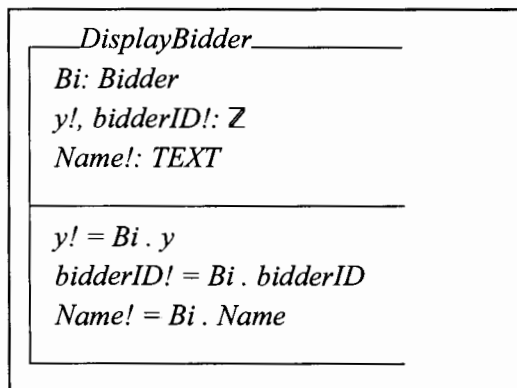
Schema Name: Round_AM

Description: Round_AM is the state schema made to store computed round keys, so that AM can access the computed Round keys by using the function ComputeRK and store these values in a sequence BInfo. This is done for maintaining the anonymity because if AM accesses the round key from the RoundKeyBoard; then the confidentiality of the bidder's public key cannot be maintained.

4.3 Dynamic Formal Model

The dynamic model consists of all operational schemas of the English Auction Protocol.

4.3.1 Display Bidder



Schema Name: DisplayBidder

Description: DisplayBidder schema is an operational schema consisting of four components, and changes the state of the Bidder, participant of the auction. 'Bi' of type Bidder is used to access the parameters from Bidder schema. *y!* and *bidderID!* are output integers that are used for displaying the bidder id, public key and the name of the bidder as NAME! of type text.

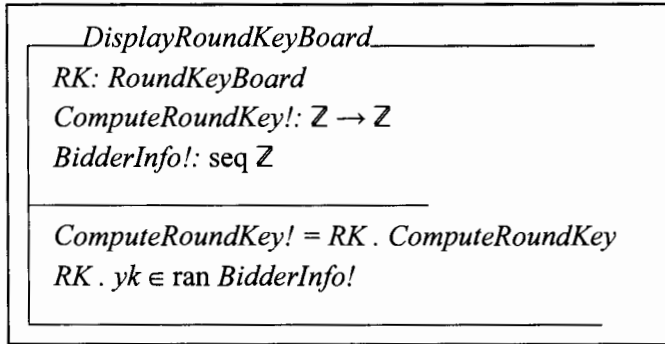
Pre Condition:

- The pre condition is true by default in this schema.

Post Condition:

- This schema displays the the bidder's public key, bidder's Identity and the name of the bidder.

4.3.2 Display Round Key Board



Schema Name: DisplayRoundKeyBoard

Description: DisplayRoundKeyBoard operation changes the state of the RoundKeyBoard schema. Already computed Round Keys will be displayed using this operation and new Round keys for new bidders will be calculated after exhibiting these keys. This operation uses ‘ComputeRoundKey!’ function to display the Round keys computed on Round Key Board schema. These Round Keys then will be stored in the BidderInfo! sequence.

Pre Condition:

- The pre condition is true by default in this schema.

Post Condition:

- This schema displays the report, the Round keys of the bidders and these Round key must belong to the range of sequence BidderInfo! in which all the Round keys have been stored already.

4.3.3 Display Auction Manager's Board

<p><i>DisplayAMBoard</i></p> <p><i>AMB: AuctionManagerBoard</i></p> <p><i>Ti!, RandomNo!, G!: Z</i></p> <p><i>Tymapping!: Z → Z</i></p> <p><i>TGmapping!: Z → Z</i></p> <hr/> <p><i>Ti! = AMB . Ti</i></p> <p><i>RandomNo! = AMB . RandomNo</i></p> <p><i>G! = AMB . G</i></p> <p><i>Tymapping! = AMB . Tymapping</i></p> <p><i>TGmapping! = AMB . TGmapping</i></p> <p>$\forall Tj: Z$</p> <ul style="list-style-type: none"> $(Ti!, RandomNo!) \in Tymapping! \wedge (Tj, RandomNo!) \in Tymapping! \Rightarrow Ti! = Tj$ <p>$\forall Tj: Z \cdot (Ti!, G!) \in TGmapping! \wedge (Tj, G!) \in TGmapping! \Rightarrow Ti! = Tj$</p>

Schema Name: DisplayAMBoard

Description: The DisplayAMBoard is used to display the values for other's verification. The DisplayAMBoard operation changes the state of the AuctionManagerBoard schema by using AMB of type AuctionManagerBoard in order to display the auction tickets generated for each round key of a bidder. The output variables $Ti!$, $RandomNo!$, and $G!$ are output integers. The outputs are displayed by getting the values from Auction Manager Board using AMB of type Auction Manager Board. $Tymapping!$ and $TGmapping!$ are the output functions used to show the correspondence of $RandomNo!$, $Ti!$ and $G!$ of the particular bidder.

Pre Condition:

- The pre condition is true by default in this schema.

Post Condition:

- This schema displays the report about the Auction Ticket, the shared parameter of bidder and the Auction Manager 'RandomNo', the element 'G' provided by

the Auction Manager and the correspondence of the bidder's Auction Ticket, the

RandomNo and the element G.

4.3.4 Generate Round Key

<i>GenerateRoundKey</i>
$\Delta Bidder$
$\Delta RoundKeyBoard$
$\Delta AuctionManagerBoard$
$\Delta BiddingBoard$
$ComputeRoundKey \in F(\mathbb{Z} \times \mathbb{Z})$
$ComputeRoundKey' \in F(\mathbb{Z} \times \mathbb{Z})$
$Registrationboard \neq \emptyset$
$\vee \# Registrationboard < maxsize \vee ComputeRoundKey = \emptyset$
$\vee \# ComputeRoundKey > 0 \wedge bidderID \notin \text{ran } BidderInfo \vee bidderID \in \text{ran } BidderInfo$
$\vee Tymapping = \emptyset \wedge TGmapping = \emptyset \mid \vee TMmapping = \emptyset \wedge TVmapping = \emptyset$
$\# ComputeRoundKey' > 0 \wedge bidderID' \notin \text{ran } BidderInfo' \vee bidderID' \in \text{ran } BidderInfo'$
$\vee Tymapping' = \emptyset \wedge TGmapping' = \emptyset \vee TMmapping' = \emptyset \wedge TVmapping' = \emptyset$
$ComputeRoundKey' = ComputeRoundKey \cup \{(Bi . bidderID \mapsto yk)\}$

Schema Name: GenerateRoundKey

Description: The operation schema GenerateRoundkey brings a change in Bidder as bidder's parameters will be passed to the Auction Manager after the generation of the round keys. Round keys are evaluated on the basis of the components taken from RoundkeyBoard. The round keys are generated by the Registration Manager for the bidders. So that bidders might get registered to the auction. This is the first step of the randomization operation in Round set up process. AuctionManagerBoard is then passed to the Bidding Board as bidders are partially valid bidders now.

Pre Conditions:

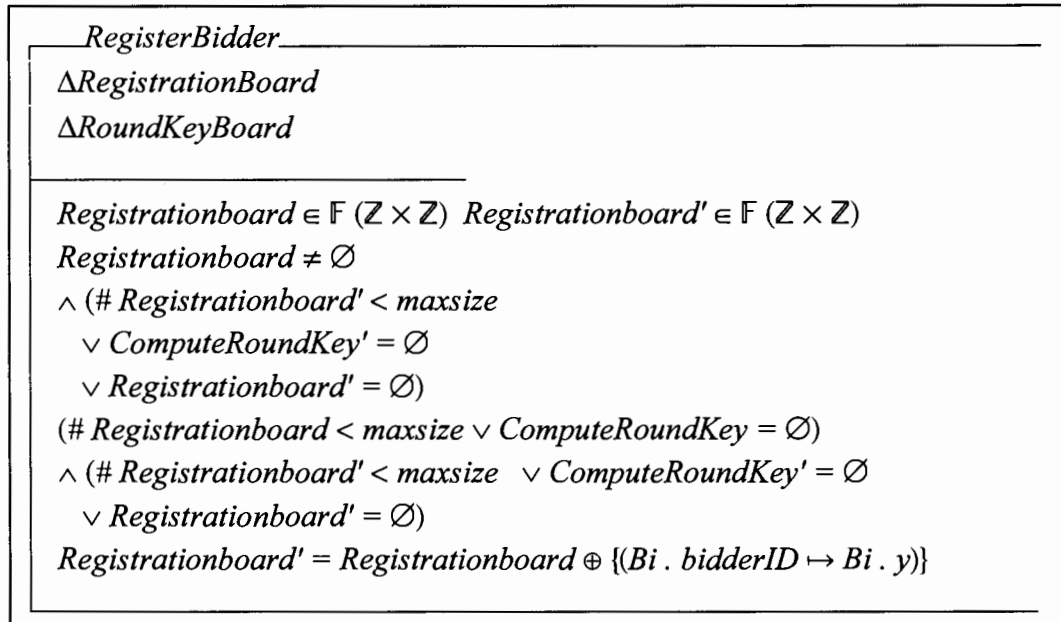
- Registrationboard function is initially empty which compares the bidderID with the public key of the bidder.
- Registrationboard function must be less than maxsize.

- ComputeRoundKey function is empty initially in which the bidders public key 'y' and hash function with bidder's variable 'ti' has been used.
- Number of Round Keys must be greater than '0' and bidderID! must not belong to range of BidderInfo that are the Round keys.
- BidderID! must belong to the Round Keys as range of BidderInfo.
- Tymapping! and TGmapping! are empty initially in which there is a correspondence among the bidder's values RandomNo!, Ti! and G!.
- TVmapping! and TMmapping! are empty in which there is a correspondence among the values Ti!, V₁! and m of the particular bidder.

Post Conditions:

- Cardinality of ComputeRoundKey' must be greater than zero and bidderID!' must not belong to the range of bidderInfo' which is a sequence of the Round Keys.
- The bidder's bidderID!' must belong to the range of BidderInfo'.
- Tymapping!' and TGmapping!', the output post state functions are empty.
- TMmapping' and TVmapping', the post state functions are empty.

4.3.5 Register Bidder



Schema Name: RegisterBidder

Description: The operation RegisterBidder formulates change in RegistrationBoard as the bidder has not been registered already. Bidder initially has to register to the registration manager to participate in the auction. Bidder has got his private and public keys after the completion of the registration process and bidder can then able to participate in the auction.

Pre Conditions:

- Registration Board is not empty.
- Cardinality of Registrationboard is less than maxsize or the ComputeRoundKey function is empty.

Post Condition:

- Cardinality of RegistrationBoard' must be less than maxsize or the ComputeRoundKey' and Registrationboard' functions are empty.

4.3.6 Generate Auction Ticket

<u>GenerateAuctionTicket</u>
$\Delta AuctionManagerBoard$ $\Delta BiddingBoard$ $\Delta Round_AM$
$ComputeRK \in F(\mathbb{Z} \times \mathbb{Z})$ $ComputeRK = \emptyset$ $\vee \# ComputeRK > 0 \wedge Bi . bidderID \notin \text{ran } BInfo \vee Tmapping = \emptyset \wedge TGmapping = \emptyset$ $\vee TMmapping = \emptyset \wedge TVmapping = \emptyset$ $\vee Bi . bidderID \in \text{ran } BInfo \vee \# BidderInfos > 0 \wedge RandomNo \notin \text{ran } BInfo \vee$ $RandomNo \in \text{ran } BInfo$ $\# BidderInfos' > 0 \wedge RandomNo' \notin \text{ran } BInfo'$ $\vee RandomNo' \in \text{ran } BInfo'$ $\vee TMmapping' = \emptyset \wedge TVmapping' = \emptyset$ $BidderInfos' = BidderInfos \oplus ComputeRK$

Schema Name: GenerateAuctionTicket

Description: The GenerateAuctionTicket schema is using the DisplayRoundKeyBoard, AuctionManagerBoard and the BiddingBoard schema to ensure the correctness of the pre and post conditions of the schema GenerateAuctionTicket. This operation brings a change in the Round as already computed round keys are used by AM.

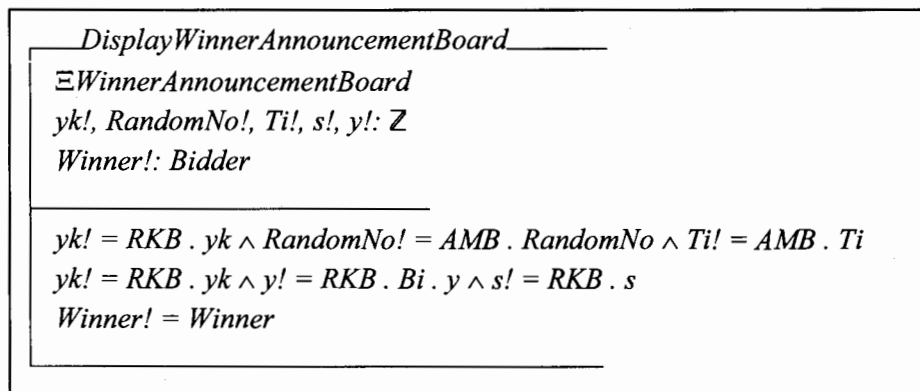
Pre Conditions:

- Cardinality of the ComputeRoundKey! must be greater than '0' and bidderID must not belong to the range of BidderInfo!.
- Tymapping and TGmapping functions are empty initially in which there is a correspondence among the bidder's values RandomNo, Ti and G.
- TVmapping and TMmapping are empty in which there is a correspondence among the values Ti , V₁ and m of the particular bidder.
- The bidder's identity bidderID must belong to the range of BidderInfo!, the range of Round Keys.
- The cardinality of the pre state BidderInfos must be greater than zero and the RandomNo must not belong to the range of BidderInfo!.
- The RandomNo of the bidder must belong to the range of BidderInfo!, that is the RoundKeys.

Post Conditions:

- The cardinality of the #BidderInfos' must be greater than '0' and the post state of RandomNo' must not belong to the range BidderInfo!'.
- The bidder's RandomNo' must belong to the range of BidderInfo!'.
- TMmapping' and TVmapping' are empty.

4.3.7 Display Winner Announcement Board



Schema Name: DisplayWinnerAnnouncementBoard

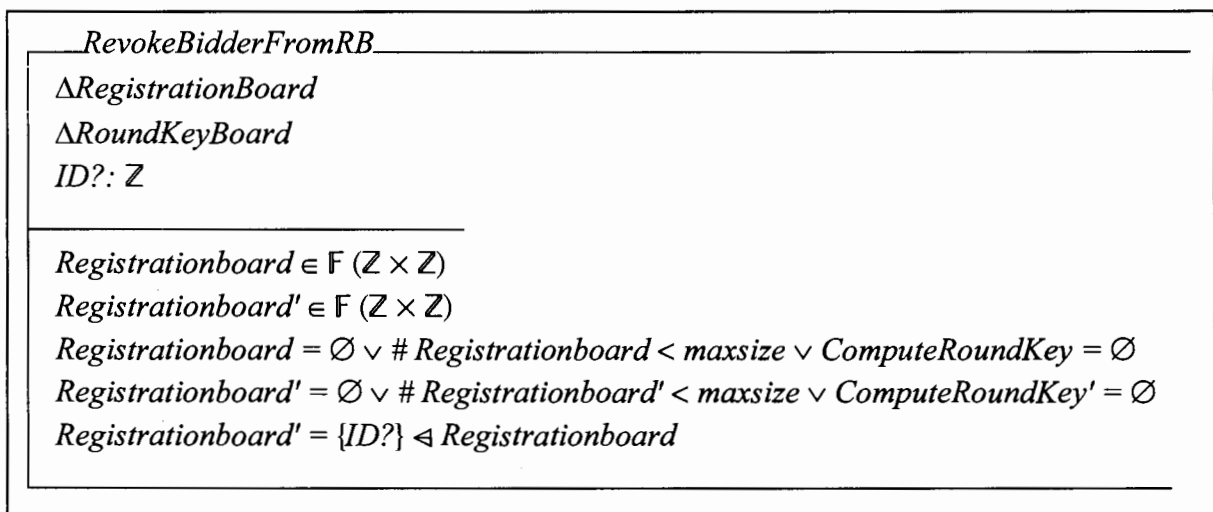
Description: The DisplayWinnerAnnouncementBoard schema has been designed to display the values for the Winner Announcement Board that will be needed for the verification. The five output integers $yk!$, $RandomNo!$, $Ti!$, $s!$ and $y!$ are used to display the bidder's values. The $Winner!$ is bidder from Bidder schema. The bidder's Round key from RoundKeyBoard has been displayed through $yk!$. The $RandomNo$ given by the Auction Manager to the bidder is displayed by $RandomNo!$ and the AuctionTicket Ti from Auction Manager Board is displayed using $Ti!$ to show the correspondence among these parameters ($yk!$, $RandomNo!$, $Ti!$) of the Bidder. The other demonstration of the bidder's parameters is $yk!$ from Round key board, $y!$, the public value of the bidder and $s!$ is the encrypting parameter hash function including bidder's value ti . $Winner!$ is the bidder identified from the Winner Announcement Board.

Pre Condition:

- The pre condition is true by default in this schema.

Post Conditions:

- This schema displays the report in different chunks like firstly there is a display of the Round key of the bidder from the Round Key Board along with the Random No from the Auction Manager Board and the Auction Ticket by the Auction Manager of the bidder.
- The winner will be displayed from the Winner Announcement Board.

4.3.8 Revoke Bidder from Registration Board

Schema Name: ReveokeBidderFromRB

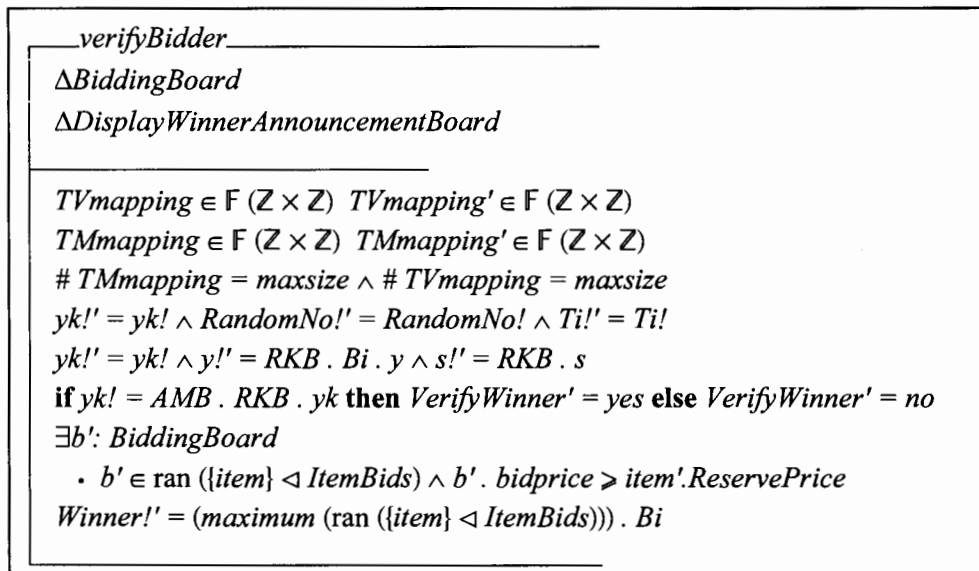
Description: The RevokeBidderFromRB comprises of RegistrationBoard and DisplayRoundKeyBoard which will satisfy the pre and post conditions of the Revocation of the bidder that wants to quit after registration from the Registration Board schema.

Pre Condition:

- Registrationboard function is empty initially or the cardinality of the Registrationboard must be less than maxsize or the output type function ComputeRoundKey! must be empty.

Post Conditions:

- Cardinality of the Registrationboard' must be less than maxsize
- ComputeRoundKey!' is empty.

4.3.9 Verify Bidder**Schema Name:** verifyBidder

Description: VerifyBidder operation does the change in WinnerAnnouncementBoard and BiddingBoard as there is a need to compare the bidder's parameters along with the values present in the WinnerAnnouncementBoard and the comparison of bid values from the BiddingBoard.

Pre Condition:

- The cardinality of the TMmapping is equal to the maxsize and the cardinality of the TVmapping must be equal to maxsize in which T_i , m and V_i are published of a unique bidder.

Post Conditions:

- The bidder's Round Key $yk!$, Random No! and $T_i!$ are displayed.
- The bidder's Round Key $yk!$, public key $y!$ and the hash function with t_i of bidder as $s!$ are updated.
- The bidder's $b!$ must belong to the range of items and the bidprice of the bid must be greater than the item's Reserve price.
- The Winner! must be the maximum among the number of bids from ItemBids function.

4.3.10 Revocation of Bidder from Round Key Board*RevokeBidderFromRKB* Δ RegistrationBoard Δ RoundKeyBoard Δ AuctionManagerBoard Δ BidderID?: \mathbb{Z} $Registrationboard \in F(\mathbb{Z} \times \mathbb{Z}) \mid Registrationboard' \in F(\mathbb{Z} \times \mathbb{Z})$ $ComputeRoundKey \in F(\mathbb{Z} \times \mathbb{Z}) \mid ComputeRoundKey' \in F(\mathbb{Z} \times \mathbb{Z})$ $Registrationboard = \emptyset \vee \# Registrationboard < maxsize \vee ComputeRoundKey = \emptyset$ $\vee \# ComputeRoundKey > 0 \wedge bidderID \notin \text{ran } BidderInfo$ $\vee Tymapping = \emptyset \wedge TGmapping = \emptyset \vee TMmapping = \emptyset \wedge TVmapping = \emptyset$ $\vee bidderID \in \text{ran } BidderInfo$ $ComputeRoundKey' = \emptyset \vee \# ComputeRoundKey' > 0 \wedge bidderID' \notin \text{ran } BidderInfo'$ $\vee bidderID' \in \text{ran } BidderInfo' \vee Tymapping' = \emptyset \wedge TGmapping' = \emptyset \vee TMmapping' = \emptyset$ $\wedge TVmapping' = \emptyset$ $ComputeRoundKey' = \{ID?\} \triangleleft ComputeRoundKey$

Description: If the bidder wants to revoke from the Round Key Board then there are these pre and post a condition that needs to be satisfied as a prerequisite. For this purpose some parameters used from the DisplayAMBoard, DisplayBidder, RegistrationBoard, BiddingBoard and DisplayRoundKeyBoard schemas.

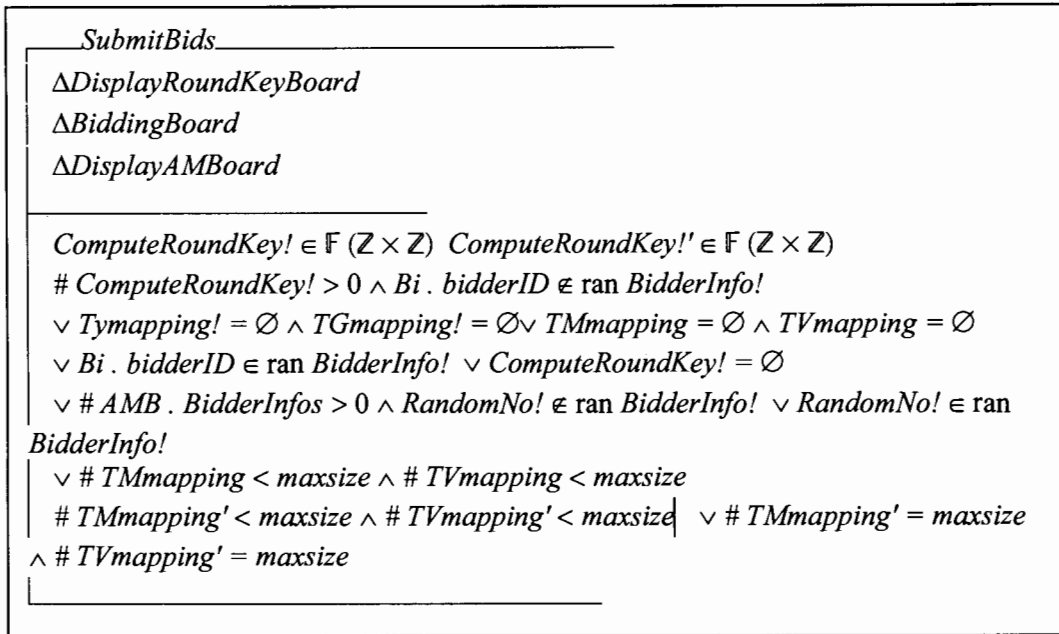
Pre Conditions:

- Registrationboard function that contains bidderID along with the bidder's public key y is empty.
- The cardinality of the Registrationboard must be less than maxsize.
- The output function ComputeRoundKey! must be greater than '0' and the bidderID! must not belong to the range of BidderInfo!.
- Tymapping! and TGmapping! are equal to phi.
- TMmapping and TVmapping are also empty.
- The output variable bidderID! must also belong to the range of BidderInfo!.

Post Conditions:

- The post state output function ComputeRoundKey!' must be empty.
- The cardinality of post state function ComputeRoundKey!' must be greater than '0' and bidderID!' must not belong to the range of BidderInfo!'.
- BidderID!' must belong to the range of BidderInfo!'.
- Tymapping!' and TGmapping!' functions are empty.
- TMmapping' and TVmapping' functions are empty.

4.3.11 Submit Bids



Schema Name: SubmitBids

Description: The SubmitBids operation changes the RegistrationBoard, RoundKeyBoard, BiddingBoard, AuctionManagerBoard and Bidder for the sake of satisfying the pre and post conditions. This schema actually verifies bids by evaluating parameters from RegistrationBoard, RoundKeyBoard, BiddingBoard, AuctionManagerBoard and Bidder. All the parameters including auction ticket, round key, signature and bid value of a particular bidder are gathered at one place to check the consistency and then the bid will be submitted.

Pre Conditions:

- The cardinality of the output function ComputeRoundKey! must be greater than '0' and the bidderID must not belong to the range of BidderInfo!.
- The output functions Tymapping! and TGmapping! are empty.
- The functions of BiddingBoard schema TMmapping and TVmapping are empty.
- The bidderID from the Bidder schema must belong to the range of BidderInfo!, that are the RoundKeys of the bidder provided by the Registration Manager.

- The output function `ComputeRoundKey!` is empty.
- The cardinality of `BidderInfos` that are the Auction Tickets provided by the Auction Manager to the bidders must be greater than '0' and the output variable `RandomNo!` must not belong to the range of `BidderInfo!`.
- The `RandomNo!` must belong to the range of `BidderInfo!`.
- The cardinality of `TMmapping` and `TVmapping` must be less than `maxsize`.

Post Conditions:

- The cardinality of `TMmapping'` and `TVmapping'` must be less than `maxsize`.
- The cardinality of `TMmapping'` and `TVmapping'` can be equal to `maxsize`.

5

Evaluation



5. EVALUATION

One of the main reasons why companies do not engage in electronic commerce is the lack of trust in e-commerce approaches which is often uttered as concerns about security (Meisels & Saaltink, 1995). There are security mechanisms that can ensure confidentiality (e.g. through encryption mechanisms such as Elliptic curve crypto system (Chen, 2004)), authenticity and non-repudiation (e.g. through digital signatures together with certificates (Omote & Miyaji, 2001), Group signature scheme (Omote & Miyaji, 2002)) Signature of Knowledge (SK) (Omote & Miyaji, 2001), and integrity of data (e.g. through digital signatures or hash algorithms (Lee et al. (2001))).

An efficient public auction protocol such as English Auction protocol by Lee et al. (2001) includes the security features as well. These informally presented properties incorporate such security mechanisms (Chen, 2004; David & Schwartz, 2003; Lee et al., 2001; Omote & Miyaji, 2001) and are formally specified in our work using Z-Notation.

5.1 EVALUATION OF THE PROPERTIES OF ENGLISH AUCTION PROTOCOL

The following section discusses the properties defined by Lee et al. (2001) along with the evaluation in context of the formalization of this study. These security properties are formally specified but their verification is informal. The verification is done by checking the conformance of the properties with the specifications of EAP. The formalization approaches for English Auction Protocol discussed in chapter 2 are now being evaluated for the approach being used for this project as shown in table 5.1.

Anonymity:- Anonymity is defined as “*Bidder's bid can't be identified by anyone in the public auction scheme.*”

- Anonymity for RM: RM cannot identify BidderID in Bidder schema from the Auction Tickets $(T_i, (Y_i^k)^{r_i}, g^{r_i})$ published by AM in AuctionManagerBoard schema used in GenerateAuctionTicket operation schema. RM also fails to identify the bidding info (T_i, m, V_i) posted by bidder in the BiddingBoard schema. Identifying Round key Y_i^k generated in RoundkeyBoard schema used by GenerateRoundkey operation schema is a discrete logarithm problem. RM cannot identify BidderID in Bidder schema from T_i in AuctionManagerBoard schema without knowing the shared key y_A between Bidder and AM. SK (Signature of Knowledge) is used here to ensure the BidderID's confidentiality in Bidder schema as it cannot be identified from V_i in BiddingBoard schema.
- Anonymity for AM: The identity of the bidder cannot be determined through Auction Ticket. Without the knowledge of t_i in Bidder schema, AM cannot identify BidderID in Bidder schema from the round key Y_i^k in GenerateRoundkey published by RM. The identification of y_i in Bidder schema from Y_i^k (Round key) in GenerateRoundkey operation schema is a discrete logarithm problem. AM can't be able to compute $s = h^k(t_i)$ that is calculated in RoundkeyBoard schema, because of cryptographic hash function. BidderId in Bidder schema cannot be identified from V_i in BiddingBoard schema because of SK.
- Anonymity for Bidders: Bidders can access the information which is posted on the bulletin boards. All other information is anonymous to the bidders.

It is assumed that collusion among RM and AM is not possible to break bidder's anonymity. Collusion may cause identification of any bidder. Public way cooperation is only made for winning bid.

Traceability:- Traceability determines "Who is the winning bidder at the end of bidding process". The winning bidder must be identified at the end of the auction. So that winner's identity can be displayed in DisplayWinnerAnnouncementBoard schema by publishing the RandomNo! from AuctionManagerBoard schema and $s!$ equivalent to $h^k(t_i)$ from RoundkeyBoard schema. A winner's identity can be identified with the cooperation of AuctionManagerBoard schema and RoundkeyBoard schema together as shown in DisplayWinnerAnnouncementBoard schema.

Table 5.1 Comparison of the Properties of English Auction Protocol

Author	Dumas et al. [2001]	Chen et al. [2003]	Cheng et al. [2006]	Cheremisinov et al. [2006]	Badic a et al. [2008]	Our Approach
Language used	DeLp	LOTOS	LTL	PRALU	FSP	Z
Security Properties						
➤ Anonymity	No	No	No	No	No	Yes
➤ Traceability	Yes	No	Yes	No	No	Yes
➤ No framing & Unforgeability	No	No	No	No	No	Yes
➤ Unlinkability in different rounds of auction	No	No	No	No	No	Yes
➤ Linkability in a round of auction	No	No	No	No	No	Yes
➤ Fairness	No	No	No	No	No	No
➤ Easy revocation	No	Yes	No	Yes	Yes	Yes
➤ Verifiability	No	Yes	Yes	No	Yes	Yes
➤ One time registration	No	No	No	Yes	No	Yes

No Framing & Unforgeability:- No framing can be defined as “*The security against framing attacks such that an entity impersonates another valid bidder.*” BidderId in Bidder schema can’t be impersonated by RM and AM because V_i , the signature of a bidder, is computed as $h^k(t_i)x_i$ in BiddingBoard schema. Since, the value of x_i is only known to bidder, present in Bidder_PrivateKey schema, so the collusion of RM and AM cannot impersonate BidderId in the Bidder schema. Unforgeability is defined as “*No one can forge a bid with valid signature.*” BidderID cannot be forged by anyone including

RM and AM with a signature V_i in BiddingBoard schema. This feature is attained by encrypting private key 'x' with the bidder's bid 'm' using Encrypt axiomatic definition. The signature V_i is providing the facility of No framing and Unforgeability as private key 'x' is used in it which is only known to the bidder and 'ti' is also a random number chosen by the bidder in order to evaluate the signature.

No linkage among plural Rounds of Auctions:- It is defined as "*No one is in a position to draw up conclusions regarding the linkability of individual bidders during different auction rounds.*" The Auction Ticket 'Ti' is generated by two randomization operations: RM (RoundKey Generation schema in GenerateRoundkey schema) on the Round Key Board and the Auction ticket generation 'Ti' by AM in AuctionTicketGeneration schema, so the Auction Ticket 'Ti' cannot be linked to a bidder's public key 'y' or the bidder's identity. Hence, bidder's bid among plural rounds of auctions cannot be linked.

Linkage in a Round of auction:- It is defined as "*Anybody can link which bids are placed by same bidder and knows how many times a bidder places a bid in a round of auction.*" In a given auction round, anyone can determine the number of bids made by a bidder, and tell the bidding price is submitted by which bidder. The same Auction Ticket 'Ti' is used in a round of Auction. This makes it easier for anyone to link from the SubmitBid schema that;

- How many times the same bidder has placed a bid (m, T_i, V_1) where m is the bid of bidder, T_i is the auction ticket and V_1 is the signature of the bidder.
- Which bids are placed by the same bidder in a round of auction.

Fairness:- *Fairness is defined as "All the bidders are dealt in a fair way."* Fairness in the process can be achieved because the Bulletin boards are used by AM and RM, therefore the behavior of the managers will turn out on the Bulletin Boards and any bidder can point out any misbehavior. Fairness is not precisely defined by Lee et al. (2001) in his work. As this parameter is ambiguously defined, so it is ignored in this work.

Easy Withdrawal:- It is defined as “*Registration managers can easily and efficiently revoke the bidding rights of a bidder.*” When a bidder wants to withdraw from an auction or RM wants to revoke a certain bidder, a revocation of bidder can be conducted in a simple and easy manner. It is uncomplicated to revoke a bidder because revocation can be done by removing the Round key of the particular bidder from the RoundKeyBoard schema and Bidder from the RegistrationBoard schema.

Verifiability:- Verifiability determines “Whether or not a bid is coming from a valid bidder, i.e., validity of the bidder.” Anybody should be able to verify a signature on a bid and confirm whether the bidder is valid or not. Further the bidder, bid price validity, and the winning bidder should be verifiable. Since relevant information is posted in different schemas, bid validity can be determined through the signature of the Bidder V_i , calculated in BiddingBoard schema. Verification of the signature ‘ V_i ’ on a bid is declared in the Bidder schema. This value is evaluated by using ‘ x ’ (private integer key of the bidder) and is encrypted through Encrypt function in the BiddingBoard schema of the bidder. The validity of the bidder can be validated by round key ‘ Y_i^k ’ from GenerateRoundkey schema and auction ticket ‘ T_i ’ from GenerateAuctionTicket schema. Legitimacy of bidder can be verified by checking auction key in AM, correctness of the winner announcement using ‘ $ri!$ ’ and ‘ $s!$ ’ by Auction Manager on the DisplayWinnerAnnouncementBoard schema.

One-time registration:- It is defined as “*Bidder participates anonymously with one time registration in plural rounds of auction.*” A bidder needs to register only once, after which he can participate in plural rounds of auction. The ‘winner!’ is displayed in DisplayWinnerAnnouncementBoard schema in a round of auction. The anonymity of the Auction Ticket ‘ T_i ’ in AuctionTicketGeneration schema, is maintained in the subsequent rounds, as there are different Round keys in GenerateRoundkey schema for each round. Because of this, Auction Ticket is also different for different rounds. Therefore, bidders can take part anonymously in plural rounds of auction with one time registration as a valid bidder.

The problems can still arise, if the same bidder registers two times with different private keys? This kind of problem usually is categorized as shilling behavior, and many works have already been done to handle these kinds of problems. These works can be used to resolve the above mentioned problem.

If we analyze the mentioned approaches in the table 5.1, we notice that out of the five approaches Chen et al. (2003) and Cheremisinov & Cheremisinova (2006) are the two approaches that can be considered the best as they address four out of twelve parameters of English Auction Protocol. The proposed approach, using Z specification language, is better than the existing approaches since it addresses almost all the key security parameters. This is so because we have taken into account the security aspect of English Auction Protocol as described by Lee et al. (2001).

6

Analysis of the Proposed Model

6. ANALYSIS OF THE PROPOSED MODEL

The resultant formal model derived from Finite State Automata of English Auction Protocol is now ready to be validated by a model checker which is Z/EVES tool set. Z/EVES is a tool for analyzing Z specifications. It can be used for parsing, type checking, domain checking, schema expansion, precondition calculation, refinement proofs, and proving theorems (Monin, 2003).

6.1 Analyzing Z Specifications

The use of Z, or any other formal notation, in a specification is a great step, as natural language specifications are notorious for their ambiguity. Making formal statements about a system can itself be beneficial, as it can lead to a careful consideration of the important aspects of the system and to the development of a consistent terminology (Bowen, 1997).

However, if formal specifications of a system have been developed, it does not mean that the specifications are correct and consistent. There can be different types of errors ranging from trivial spelling errors to inconsistencies in meaning.

The significant feature of formal specifications which excel all other traditional means of informal specification is that formal specifications can be checked and analyzed for the presence of errors. It allows proving various properties of the system by proving the specifications using various exploration techniques. Since Z/EVES is a powerful tool, it supports several means to explore the formal specifications. Some of exploration techniques offered by Z/EVES have been used to analyze the formal English Auction Protocol. The techniques given by Meisels & Saaltink, 1997) are presented as follows.

6.1.1 ERROR CHECKING

i) Syntax and type checking: The Z language has quite complex syntax, and the

Mathematical Toolkit contains number of functions. It is easy for an inexperienced Z user, to make a mistake while developing the specifications. So, Z/EVES detect and report such type errors. Z/EVES can be used incrementally as each paragraph of a specification is written, it can be immediately checked and, if necessary, corrected unlike other tools (Meisels & Saaltink, 1997). The syntax and type checking of the static model of EAP which includes all the state schemas like Bidder, Bidder_PrivateKey, RoundKeyBoard, Auction Manager Board, Registration Board, Bidding Board, Item, and Winner Announcement Board has been done by using Z/EVES tool-set and is shown in table 6.1. Similarly the syntax and type checking of the dynamic model of EAP having all the operation schemas like DisplayBidder, DisplayBiddingBoard, DisplayRoundKeyBoard, DisplayWinnerAnnouncement Board, DisplayAMBoard, GenerateRoundKey, RegisterBidder, GenerateAuctionTicket, RevokeBidderfromRB, VerifyBidder, RevokeBidderfromRKB and SubmitBids are shown in table 6.2.

ii) Domain checking: Domain checking performs checking that is beyond the scope of syntax and type checking. In Z notation, the expressions can be written which are not necessarily meaningful. There are two types of such expressions, a function can be applied outside its domain, e.g., $1 \text{ div } 0$ or $\#N$ and the other is a definite description (μ -term) which is not meaningful if there is not a single value satisfying the predicate. e.g., $\mu x : Z \mid x \neq x$, for which there is no possible value of x satisfying the predicate, and $\mu x : Z \mid x > 0$, for which there are many possible values. The way of catering such problems in Z/EVES is to show that every expression is meaningful. Using domain analysis technique Z/EVES examines each paragraph as it is entered, and checks each function application and definite description for meaningfulness (Meisels & Saaltink, 1997). The domain checking of the static formal model containing all the state schemas and dynamic formal model which consists of all the operation schemas of EAP, is done by using Z/EVES tool-set as illustrated in table 6.3 and table 6.4 respectively.

iii) Reduction: The reduction commands traverse the current goal, accumulating assumptions and performing reduction on predicates and expressions in the goal. In a traversal, each formula and sub-formula of the goal is examined, and may be replaced by

a logically equivalent formula that Z/EVES considers "simpler". Other replacements may occur, depending on the type of reduction being performed (Meisels & Saaltink, 1997). Reduction has been performed on predicates and expressions of the static and dynamic formal model of EAP. Validation using reduction technique is shown in table 6.3 and table 6.4 respectively.

iv) Prove By Reduce: It is one of the techniques for analyzing Z specifications. The 'prove by reduce' performs repeated reducing on the current goal, until reduction has no effect. Therefore, this technique works in iterations. This command validates that the current goal is rearranged, Equality substitution is performed and the current goal is reduced for each iteration (Meisels & Saaltink, 1997). This technique has been applied on the static and dynamic formal model of EAP by using Z/EVES tool-set. Prove by reduce validation is shown in table 6.3 and table 6.4 correspondingly.

6.2 Validation of Formal English Auction Protocol

The formal English Auction Protocol is checked and analyzed against the Z/EVES Toolset. The outcome of analyzing formal English Auction Protocol is described in table 6.1 and 6.2. The schemas are analyzed with four techniques of Z/EVES tool-set explained above. Some schemas proved automatically by using prove assistance of the tool.

Table 6.1 Results of exploration of Static Model of EAP

Static Model	Syntax & type Checking	Domain Checking	Reduction	Prove by Reduce
Bidder	✓	✓	✓	✓
Bidder_PrivateKey	✓	✓	✓	✓
RoundKeyBoard	✓	✓	✓	✓
Auction Manager Board	✓	✓	✓	✓
Registration Board	✓	✓	✓	✓
Bidding Board	✓	✓	✓	✓
Item	✓	✓	✓	✓
Winner Announcement Board	✓	✓	✓	✓

Table 6.2 Results of exploration of Dynamic Model of EAP

Dynamic Model	Syntax & type Checking	Domain Checking	Reduction	Prove by Reduce
DisplayBidder	✓	✓	✓	✓
DisplayBiddingBoard	✓	✓	✓	✓
DisplayRoundKeyBoard	✓	✓	✓	✓
DisplayWinnerAnnouncement Board	✓	✓	✓	✓
DisplayAMBoard	✓	✓	✓	✓
GenerateRoundKey	✓	✓	✓	✓
RegisterBidder	✓	✓	✓	✓
GenerateAuctionTicket	✓	✓	✓	✓
RevokeBidderfromRB	✓	✓	✓	✓
VerifyBidder	✓	✓	✓	✓
RevokeBidderfromRKB	✓	✓	✓	✓
SubmitBids	✓	✓	✓	✓

The Proof checking is done by the Z/EVES tool-set as shown in figure 6.1 and figure 6.2. The Proof window provides three functions: inspection and modification of a proof script; interactive construction of a proof; and proof browsing. A proof script is a sequence of Z/EVES prover commands. These commands are displayed along with their status, which indicates whether the command has been executed or not as shown in figure 6.1 and figure 6.2. Figure 6.1 shows the proof checking of the GenerateRoundkey schema by using the 'Reduce' validation technique. 'Prove by Reduce' validation technique is used in the proof checking of RegisterBidder schema as shown in figure 6.2. Like paragraphs within a specification, individual steps can be inserted, deleted, moved, or modified. If a proof script is active, its steps can be executed and the proof can be browsed (Saaltink, 1999). Some sample proof window snapshots of dynamic schemas are attached in the figure 6.1 & figure 6.2.

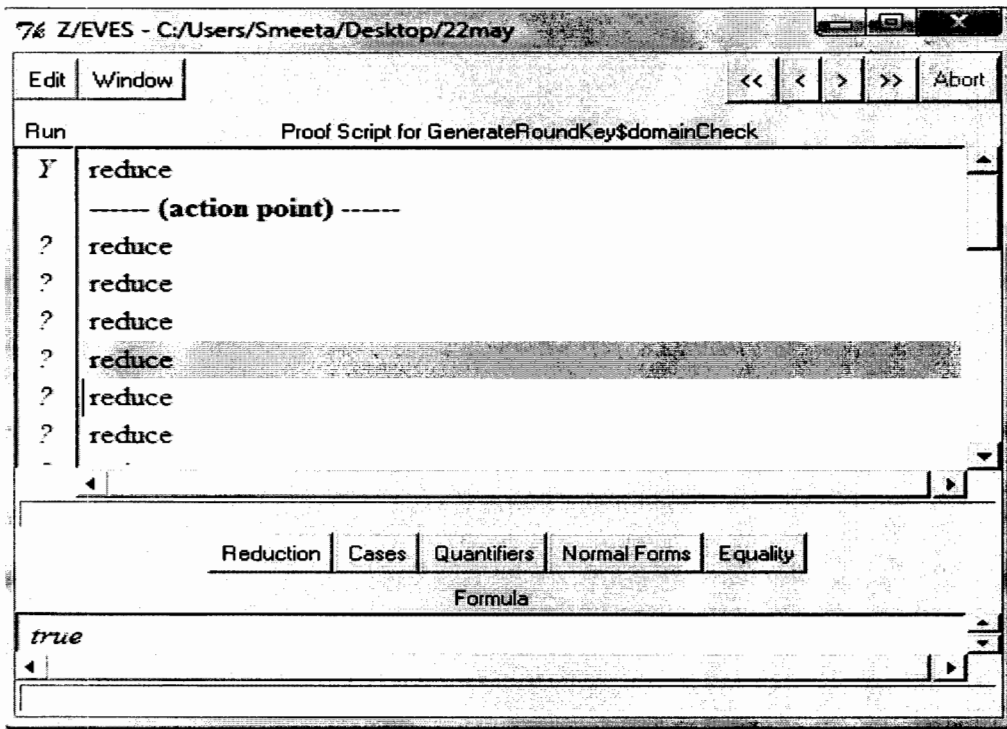


Figure 6.1 Snapshot of proof window of GenerateRoundKey schema

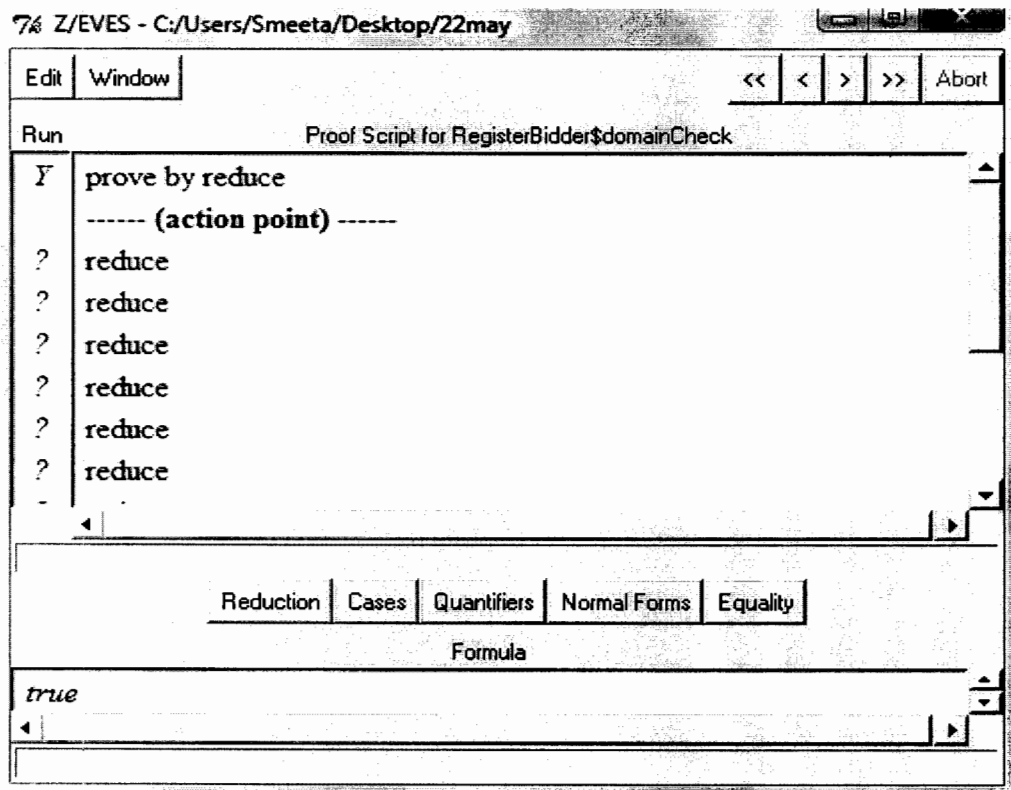


Figure 6.2 Snapshot of proof window of RegisterBidder schema

7

Conclusion and Future Works

7. CONCLUSION

The aim of this thesis was to contribute towards the effort of making the use of formal methods common in industry for the development of open auction systems in which bidding is open. Integration of automata and Z has been used to comprehend and represent multi-agent interaction systems. With the ever increasing popularity of public internet auctions emerges with the need of secure interaction protocols. The evaluation criteria adopted is the analysis of security properties defined in these approaches, due to the reason that the security lapse is being faced by the established approaches.

This work aims to enhance the current technology by modeling the multi-agent requirements through automata theory of making the Non Deterministic Automata and then transforming it to Deterministic Finite Automata of an English Auction Protocol based system to Z-Notation.

By using the example of EAP, it is shown that the proposed approach groups various properties like anonymity, traceability, no framing, linkability in a round of auction, unlinkability in different rounds of auction, unforgeability, one time registration, easy revocation and verification among agents; while taking advantage of behavior handling and semantic characterization of automata.

The proposed strategy gathers maximum security features for EAP at one platform, hence it proves to be more secure, reliable and consistent formal model for EAP. The use of proper specification language such as Z ensures correctness, reliability and consistency at analysis and design phase. As capturing errors and inconsistencies at initial stages could greatly affect the time and cost spent in later stages of the design model.

The resultant formal model of EAP using FSA incorporates certain significant security features. A formal model of English Auction Protocol using Automata is presented and also ensures validation of security properties of EAP to achieve reliable and secure EAP.

An important future direction is to consider other protocols like Contract Net Protocol etc. and to develop formal model by using the similar technique.

Many efforts have been done on the improvement of English Auction Protocol. It has been observed that most of the efforts have been scattered. For example, if one of the researchers has taken shilling as a core area of internet, the other had most of his focus on concurrent behaviours. For the future improvement in English Auction Protocol, there is a need to put together all the traits and focus on bringing out the best/optimum of all the traits in to one comprehensive model.

References

- Airchinnigh, M., Butler, M. (1993). *Service specification using Z'*, Technical Report A222BR02, SCORE Project (EU RACE Project R2017), Broadcom Eireann Research Dublin, Z Manual, 11 October.
- Ambroszkiewicz, S., Komar, J. (1999). *A model of BDI agent in game theoretic framework*, ESPRIT Project ModelAge Final Workshop on Formal Models of Agents, Vol. 1760, PP. 523-526, Springer Berlin / Heidelberg.
- Badica, A., Badica, C. (2008). *Formalizing Agent Based English Auction using Finite State Process Algebra*, Journal of Universal Computer Science, Vol. 14, No. 7, Springer Berlin / Heidelberg.
- Bowen, J., Hinchey, M., Till, D. (1997). *The Z Formal Specification Notation*, 10th International Conference of Z Users Reading, LNCS: 1212, PP. 391, 3-4 April, University of Reading, England, Springer Berlin/ New York.
- Bowen, J. (2006). *The Z notation: tool support*, Retrieved April 23, 2008, from <http://vl.zuser.org/#tools>.
- Boyd, C., Mao, W. (2000). *Security Issues for Electronic Auctions*, Hewlet Packard, HP Technical Report HPL-2000-90, Information Security Research Centre Queensland University of Technology, Australia, May 12.
- Brazier, F.M.T., Dunin-Keplicz, B.M., Jeanings, N.R., Treur, J. (1997). *DESIRE: Modelling MultiAgent systems in a compositional formal framework?*, International Journal of Cooperative Information Systems, Vol. 6, PP. 67-94.
- Busoni, L., Schutter, B, Babuska, R. (2005). *Learning and coordination in dynamic multiagent systems?*, Interactive Collaborative Information Systems, Collaborative Decision Making, Technical report 05-019, October.
- Chainbi, W., Jmaiel, M. Abdelmajid, B.H. (1998). *Conception, Behavioural Semantics and Formal Specification of MultiAgent systems*, 4th Australian Workshop on Distributed Artificial Intelligence, Multi-Agent Systems: Theories, Languages, and Applications, PP. 16-28, Springer Berlin / Heidelberg.
- Chaum, D., Heyst, E. (1991). *Group signatures, Advances in Cryptology—EUROCRYPT'91*, Lecture Notes in Computer Science, Vol. 547, PP. 257–265, Springer-Verlag, Berlin.
- Chen, T. (2004). *An English auction scheme in the online transaction environment*, Journal Computers & Security , Vol. 23, No. 5, PP. 389-399, Elsevier, July.

- Chen, B., Sadaoui, S. (2003). *Simulation and Verification of a Dynamic Online Auction*, Proceedings of 7th International Conference on Software Engineering & Applications, November 3-5, PP. 385-390, Marina del Rey, CA USA.
- Cheng, Y. (2006). *A Formal Approach to Detecting Shilling Behavior in Concurrent Online Auctions*, International Journal of Modern Physics A, Vol. 2, No. 1, PP. 15-30, ICEIS.
- Cheremisinov, D., Cheremisinova, L. (2006). *Developing Agent Interaction Protocols With Pralu*", International Journal Information Theories & Applications, Vol. 13, No. 3, PP. 239-245, FOI Bulgaria.
- David, E., Schwartz, R. (2003), *Bidders' Strategy for Multi-Attribute Sequential English Auctions with a Deadline*, Second International Joint Conference on Autonomous Agents and Multiagent systems, Melbourne, Australia July 14 - 18, PP. 457- 464, ACM New York, USA.
- Decker, K.S. (1994). *TÆMS: A Framework for Analysis and Design of Coordination Mechanisms*", Foundations of Distributed Artificial Intelligence, ISBN 978-3-540-60805-9, Wiley Inter-Science.
- Dumas, M., Governatori, G., Hofstede, A., Oaks, P. (2002). *A Formal Approach to Negotiating Agents Development*, Journal of Electronic Commerce Research and Applications, Vol.1, No. 2, PP. 193-207, Elsevier Science.
- FIPA (Foundation for Intelligent Physical Agents). (2000). *FIPA Request Interaction Protocol Specification*. Foundation for Intelligent Physical Agents, Retrieved May 26, 2010 from <http://www.fipa.org/specs/fipa00031/XC00031F.pdf>
- Hilaire, V., Koukam, A., Gruer, P., Muller, J, P. (2000). *Formal Specification and prototyping of Multi Agent Systems* , Proceedings of the First International Conference on Multi-Agent Systems, Vol. 1972/2000, PP. 254-260.
- Huaikou, M., Ling, L., Chuanjiang, Y., Jijun, M., & Li, L. (2001). *Z User Studio: An Integrated Support Tool for Z Specifications*. In *Proceedings of the Eighth Asia-Pacific Software Engineering Conference APSEC'01* (pp. 437-444). Washington, DC: IEEE Computer Society.
- ISO/IEC (2002). *Information Technology -- Z Formal Specification Notation -- Syntax, Type System and Semantics*. International standard ISO/IEC 13568:2002.
- ISO/IEC (2007). *Corrigenda, Amendments and other parts of ISO/IEC 13568:2002*. International standard ISO/IEC 13568:2002/Cor 1:2007, 2007.
- Kneuper, R. (1997). *Limits of Formal Methods*", Journal Formal Aspects of Computing, PP. 379-394, Springer London.

- Lee, B., Kim, K., Ma, J. (2001). *Efficient Public Auction with One-Time Registration and Public Verifiability*, Proceedings of the Second International Conference on Cryptology in India December 16 - 20, PP. 162-174, Springer-Verlag.
- Littman, M.L., Stone. (2002). *Implicit negotiation in repeated games* , Vol. 2333, PP. 393-404, Springer Berlin / Heidelberg.
- Meisels., Saaltink, M. (1997), *The Z/EVES Reference Manual*, ISBN: TR-97-5493-03, Release date: December 1995, Latest revision date: September 1997, ORA Canada, version 1.5.
- Monin, J. (2003). *Understanding Formal Methods*", ISBN: 1852332476, 01/2003, PP. 296, Springer Verlag.
- Munira, N., & Nadeem, A. (2009). *A formal Model for English Auction Protocol*, Seventh ACIS International Conference on Software Engineering Research, Management and Applications, PP:119-126 , December 2-4, HAIKOU, China, IEEE.
- Omote, K., Miyaji, A. (2001). *A Practical English Auction with One-time Registration*, Journal of IEICE (Institute of Electronics, Information and Communication Engineers), Vol. 100, No. 694, PP. 57-62, IEICE, Japan.
- Omote, K., Miyaji, A. (2002). *A Practical English Auction with Simple Revocation*, Journal of IEICE (Institute of Electronics, Information and Communication Engineers) Trans. Fundamentals, Vol. E85-A, No. 5, May.
- Pitt, J., Guerin, F., Stergiou, C. (2000). *Protocols and Intentional Specifications of Multi-Party Agent Conversations for Brokerage and Auctions*, International conference on Autonomous agents June 03 - 07, Spain, PP. 269-276, ACM, New York, NY, USA.
- Roggero, D., Patrone, F., Mascardi. (2005). V., "Designing and implementing Electronic Auctions in a Multiagent System Environment", Proceedings of the WOA 2005 National Workshop Dagli Oggetti Agli Agenti, PP. 157-163, Pitagora Editrice Bologna.
- Saaltink, M. (1999). *The Z/EVES 2.0 User's Guide*, TR-99-5493-06a, Ottawa, Ontario K1N 7B7, Canada, October.
- Schoop, M. (2005). *A Language-Action Approach to Electronic Negotiations*, An International Journal on Communication, Information Technology and Work, Vol. 1, No. 1, PP. 62-79, Information Systems I, University of Hohenheim, Germany.
- Spivey, M., J. (1992). *The Z Notation: A Reference Manual*. Second Edition, Englewood Cliffs, New York: Prentice- Hall. ISBN 0-13-978529-9.
- Ison, T. (2008). *Yahoo! Japan Auctions Compromised*, Retrieved May 27, 2010 from <http://www.darkreading.com/security/vulnerabilities/showArticle.jhtml?articleID=21120124>

