Security Model for a Multi-Agent Marketplace

Ashutosh Jaiswal
Dept of Computer Science
and Engineering,
University of Minnesota
ashutosh@cs.umn.edu

Yongdae Kim
Dept of Computer Science
and Engineering,
University of Minnesota
kyd@cs.umn.edu

Maria Gini
Dept of Computer Science
and Engineering,
University of Minnesota
gini@cs.umn.edu

ABSTRACT

A multi-agent marketplace, MAGNET (Multi AGent Negotiation Testbed), is a promising solution to conduct online combinatorial auctions. The trust model of MAGNET is somewhat different from other on-line auction systems: the mediated marketplace is a partially-trusted third party. In this paper, we identify the security vulnerabilities of MAGNET and present a solution that overcomes these weaknesses. Our solution makes use of three different existing technologies with other standard cryptographic techniques: publish/subscribe systems that provide simple and more general messaging, time-release cryptography to provide guaranteed nondisclosure of the bids, and anonymous communication to hide the identity of the bidders until the end of the auction. By doing so, we successfully minimize the trust on the market as well as increase the security of the whole system. The protocol that we have developed can be adapted for use by other agent-based auction systems, which use a third party to mediate transactions.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; E.3 [**Data Encryption**]: Public key cryptosystems; K.4.4 [**Computers and Society**]: E-commerce—*Security*; K.6.m [**Miscellaneous**]: Security

General Terms

Design, Security

Keywords

Electronic Auctions, Multi-agent Systems, Security

1. INTRODUCTION

The business-to-business (B2B) e-commerce market is expected to expand rapidly in coming years, with the global market expected to exceed $7.29 trillion in 2004, according to Gartner Group research. Online marketplaces, as a meeting point for B2B businesses, offer benefits to both buyers and sellers. For buyers, a marketplace can significantly ease the process of searching for and comparing providers, while for sellers marketplaces provide access to much broader customer bases [10]. Sellers and buyers can maximize their interests by making use of auction-based marketplaces.

MAGNET and the supporting architecture provides support for complex agent interactions, such as in automated contracting, as well as other types of negotiation protocols [6]. MAGNET's auctions are reverse auctions, since the auctioneer pays instead of getting paid; first priced, since the bids selected are the lowest cost feasible bids; sealed bid, since the auctioneer is the only one who sees the bids; and combinatorial, since bids can include multiple tasks with a single price for the combination. Such an architecture can be used for carrying out contracting activities required in a B2B marketplace. However, in the absence of a secure architecture, its utilization in real world is still elusive.

When the original MAGNET system was designed, security was not a major concern. However, as the system has evolved, it has become clear to us that in order to be used on open networks, a security architecture needs to be in place. Specifically, MAGNET has problems with secrecy of bids, non-repudiation, early bid opening, and manipulation of bids. These problems are quiet common in auction systems. However, in MAGNET there exists a notion of a trusted third party: the market.

The presence of the market poses a unique challenge, that of ensuring the sanctity of the trust endowed in it. We came to realize that by ensuring this trust in the market we could overcome most of the existing security problems in MAGNET. Our solution is achieved by a little bit of tweaking with the market architecture itself. The market will use a publish/subscribe system to notify other agents about its actions. By cross-checking the actions taken by the market, other agents can ensure that the market is acting properly. Thus our notion of trust is dependent on the vigilance of other agents. In a similar manner, we can ensure that other agents are acting in a proper manner.

This paper is organized as follows. In Section 2 we examine the design of MAGNET and the resulting vulnerabilities. In Section 3 we present the security assumptions for MAGNET. We outline the proposed protocol in Section 4. In Section 5 we analyze the efficiency and security of the protocol that we presented. We compare our work with existing methods in Section 6. Finally, in Section 7 we present conclusions and talk about future work.

2. MAGNET AND ITS VULNERABILITIES

The motivation for coming up with a security model for MAGNET is to make it usable on public networks without compromising the data exchanged on it. MAGNET provides support for a variety of types of transactions, from simple buying and selling of goods and services to complex multi-agent negotiation of contracts with temporal and precedence constraints [5]. However, if such a system is to be used for carrying out transactions totaling billions of dollars, it is imperative that a respectable security mechanism exist in place. Such an architecture is non-existent in the current system. This section reviews the current architecture of MAGNET and the potential vulnerabilities it poses.

2.1 Current Architecture

The current architecture of MAGNET envisions the presence of mainly three entities: the customer agent, the supplier agent, and the market. The agents are self-interested agents, which attempt to gain the greatest possible profits from their endeavors. The market is the meeting point for both the customer and the supplier agents. The customer agent (also known as the Contractor agent) wants to contract with supplier agents to fulfill a set of tasks. It achieves this goal by a three step protocol, which involves the customer agent sending out Requests For Quotes (RFQs), the supplier agents responding with bids and the customer agent accepting some bid, and finally the winning supplier agent executing the bid [6]. Following is a detailed description of the three steps.

2.1.1 *Planning:*

In this phase the customer agent selects a market which specializes in certain types of product or service categories. It then comes up with its requirements and a plan which would fulfill those requirements. While coming up with the plan, the customer agent takes into account the value of its goals and the necessity of each component. Based on the plan it generates one or more RFQs and forwards them to the market.

2.1.2 *Bidding:*

In this phase the market sends notifications to the associated supplier agents about the availability of a new RFQ. Supplier agents then contact the market and download the RFQ. If the RFQ generates sufficient interest with the supplier agent, it formulates a bid in response to the RFQ. The supplier agent then forwards the bid to the market for validation and delivery to the customer agent. The market can hold the bids until the deadline of the auction is reached or forward them right away to the customer agent. On receipt of bids, the customer agent evaluates them and selects some bid, which fulfills its plan. It then sends out a bid acceptance notice to the respective supplier agent through the market. The market keeps a record of the bid acceptance and notifies the winning supplier agents of their bid acceptance and other supplier agents about their non-acceptance.

2.1.3 *Execution:*

During the execution phase, the supplier agent works on the tasks which the customer agent had accepted in the previous phase, while the customer agent monitors the execution of these tasks. The customer agent can also re-plan and issue calls for a new bid if the plan execution does not proceed according to expectations. Once the supplier agent is done executing the plan, it notifies the customer agent. The customer agent then makes a payment (or a payment commitment) to the supplier agent. This payment is recorded by the market as well. The protocol is illustrated in Figure 1.

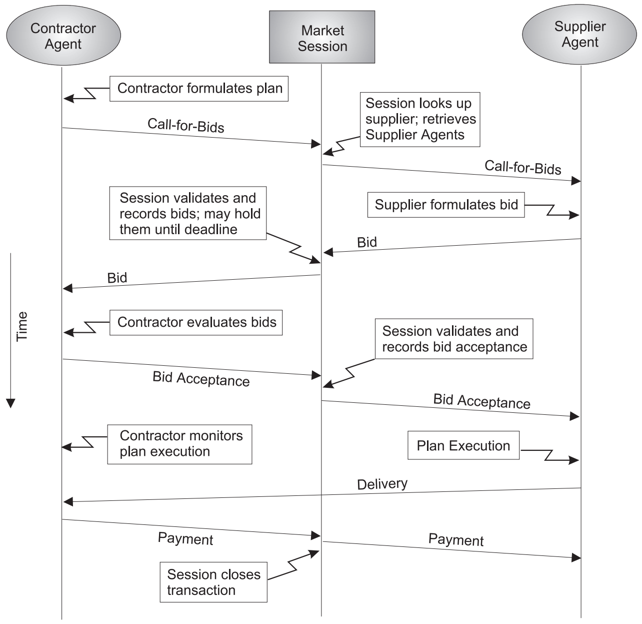
Contractor Agent
Market Session
Supplier Agent
Contractor formulates plan
Session looks up supplier; retrieves Supplier Agents
Call-for-Bids
Call-for-Bids
Session validates and records bids; may hold them until deadline
Supplier formulates bid
Bid
Bid
Time
Contractor evaluates bids
Session validates and records bid acceptance
Bid Acceptance
Bid Acceptance
Contractor monitors plan execution
Plan Execution
Delivery
Payment
Payment
Session closes transaction

Figure 1: MAGNET's original three step protocol

The planning and bidding phase of the MAGNET system have been implemented, however, the execution phase has not been implemented. Hence we would be considering the first two stages of this three-step protocol in this paper.

2.2 Vulnerabilities

The original design of MAGNET had not considered security issues. This leads to many vulnerabilities as described below.

2.2.1 *Secrecy of the bids:*

In a sealed bid auction, it is necessary that the bids are opened only after the end of the auction. The timing of the disclosure of the bid information is important. The MAGNET system is primarily designed to carry out first price, private, sealed bid, reverse, combinatorial auctions. Thus, it becomes necessary that the bid data from a supplier agent is not available to another supplier agent. In addition, MAGNET can be utilized for carrying out a public auction as well. In such a case it is imperative that the bid data is not made available, even to the customer agent before the closing time of auction. In the current system, the customer agent receives all the bids through the market. In addition none of the communication in MAGNET uses encryption of any kind at present. Thus both the customer and the market (as well as any eavesdropper) have access to the bid data before the close of the auction.

2.2.2 *Non-repudiation:*

In an auction there should be a mechanism to guarantee non-repudiation. In MAGNET, if the winning supplier agent declines to go ahead with the contract, there is no means of proving that it was indeed that agent who won the bid. Similarly, there are no means for assuring the suppliers that the RFQs they received were actually sent by the customer agent they claim to come from.

2.2.3 Prior opening of the bids:

As discussed before, the bids submitted should not be opened before the end of the auction period. MAGNET does not require that bids are not opened early, but early opening creates opportunities for counterspeculation [4]. However, an insider in the auction house can open and inform its collaborator of the contents of any bid. In MAGNET, the customer agent and the market both have access to all the bids before the end of the auction. If the customer decides to collaborate with a supplier agent, it would be harming itself, as it might lose out on a potentially better bid (see 3). However, the market can always collaborate with a supplier and give it available information pertaining to other bids. Thus the need for a mechanism to prevent early opening of bids.

Furthermore, the information about the bidder itself can be important information to the other bidders. Therefore, it is also necessary to have some kind of anonymization mechanism in the system which would make it impossible to determine the origin of a message.

2.2.4 Manipulation of closing time:

In an auction system it is possible that an insider might manipulate the closing time of an auction in order to exclude some bids from the auction. In MAGNET, the customer is free to ignore any of the bids received but it cannot extend the closing time of the auction without issuing a new RFQ. The market can, however, block new bids from suppliers to the customer agent. It can also convey the closing time differently by modifying the RFQs on their way to the suppliers.

2.2.5 Fairness:

One of the most important security requirements of an auction system is fairness: in order to maintain trust in the auction system, it is necessary that the bidders be assured that their bids were given fair treatment before deciding the winning bidder. Being a primarily private auction, fairness is not achievable by definition and is not one of the motivations behind the system. *Hence our solution will not address this problem.*

2.2.6 Fault tolerance:

Some of the above mentioned scenarios can be caused just because of the failure of the auction service or a bidding process. In the case of MAGNET, either the failure of the customer agent or the market can be responsible for the failure of the auction process. This is strictly speaking not a security hole but a problem which might lead to other security problems. We plan on addressing this issue in future.

3. SECURITY ASSUMPTIONS

The current architecture, in spite of having numerous security vulnerabilities in place, is an example of a unique approach in agent interaction. In earlier multi-agent systems, agents communicated and contracted with each other directly. In most cases these negotiations were complicated by an environment in which there was no mutual trust between the agents. MAGNET makes use of a trusted third party that could be utilized by agents to carry out transactions. Thus agents can utilize their resources towards plan execution and bidding instead of trying to negotiate in a chaotic manner. The main trust assumptions for the market are:

- It is responsible for conveying the RFQs from the contractor agent to interested supplier agents. It is also responsible for communicating the bids from the supplier agent back to the customer agent.
- It acts as a record keeper by keeping note of all the transactions and movement of RFQs and bids that take place through it. In case of dispute, the market will act as an arbitrator using saved records.
- It is responsible for aggregating statistical data from transpired auctions and making it available to interested parties at a later period of time. This data may affect the determination of the winning bidder. We assume that this statistical aggregation is performed correctly by the market. How to do this fairly and securely is one of our future concerns.

The customer agent is responsible for initiating contracts in the manner described earlier. At the same time we assume that the customer agent will not collude with any supplier agent. In case of customer-supplier collusion, the whole purpose of conducting an auction becomes useless. In such case, the customer would be only wasting time and resources in trying to contract through MAGNET (besides paying any fee that might be charged by the market for its services). Agents wishing to do so (possibly because of a preferred business relationship) can communicate directly with each other. Moreover, results of an auction can affect future auctions as well, since the statistical data gathered from an auction is made available for other agents to utilize in the future. We also assume that the customer agent communicates with the supplier agents only through the market and vice-versa. This assumption is necessary to ensure that avenues for customer-supplier collusion are discouraged and reduced. Furthermore, the market can keep records of all transactions being conducted which can be used to ensure non-repudiation. This assumption becomes especially important in case MAGNET is utilized for conducting a public auction.

4. PROPOSED ARCHITECTURE

4.1 Building Blocks

Before we explain details of the protocol, we briefly introduce the notations used in this paper:

PK_c	Public-key of a customer C
SK_c	Secret-key of a customer C
K_a	Symmetric-key a
m	Message
$E_{PK_c}(m)$	Public key encryption of m using PK_c
$D_{SK_c}(m)$	Public key decryption of m using SK_c
$S_{SK_c}(m)$	Signature of message m using SK_c
$TE(m)$	Time-release encryption of m
$hash(m)$	Hash of m
$Post(m)$	Public posting of m

We also introduce three key techniques used by us for our solution:

4.1.1 *Publish/subscribe systems:*

One of the major components that we intend to use in our protocol is a publish/subscribe or white-board system. In such a system, publishers can publish messages under certain topics. Subscribers can subscribe to topics of their interest and are notified of new postings under those topics, which they can then examine. Topics can be classified hierarchically and the message content defined in a way deemed suitable by the users. Such systems minimize message duplication. They have the added benefit of allowing anonymous postings by publishers and subscribers [12]. In our proposed architecture the market would host such a publish/subscribe system, to which both the customer and supplier agents will have access. All published messages will be signed by the originator, which can be verified by the agents accessing them.

4.1.2 *Time-lock puzzles:*

Another cryptographic technique which we wish to utilize in the system is the timed-release crypto, also known as time-lock puzzles [11]. These methods provide a way of encrypting a message such that no one can decrypt the message until a substantial amount of time has elapsed. Good time-lock puzzles prevent the use of parallel algorithms for decryption. Assume A wants to encrypt a message m with a time-lock puzzle for a period of T seconds. A picks at random two large primes p, q and computes $n = pq, \phi(n) = (p-1)(q-1)$. She then computes $t = TS$ where S is the number of squarings modulo n per second that can be performed by the solver. Then A picks a long random key k for some secure symmetric encryption scheme and encrypts m using k. Let us call the resulting cipher-text C_m. She then computes $C_k = k + a^{2^t} \bmod n$ for some random $a, 1 < a < n$. Since A knows $\phi(n)$, she can do this efficiently. The time-lock puzzle will contain (n, a, t, C_k, C_m). In order to extract m anybody would need to compute a^{2^t} and the only way to do this without knowing $\phi(n)$ is to perform t *sequential* squarings. The time delay ensured by this solution is not really absolute real time but some time period depending on the CPU power of the solver.

4.1.3 *Communication anonymizer:*

In absence of an anonymization technology, it becomes easy for an outsider, as well as an insider, to associate the bids to the bidder. This may not be an important requirement in certain auctions, but in MAGNET this is necessary to reduce market-supplier collusion and to make the supplier's bid unlinkable until the end of auction. A peer-to-peer (P2P) anonymizing network like Tarzan [8] can be used for this purpose. Tarzan achieves its anonymity with layered encryption and multihop routing. First, a host running an application that wants anonymity choses a group of hosts to form a path through the network. Next, this source-routing host establishes a tunnel using these hosts, which includes the distribution of session keys. Finally, it routes data packets through this tunnel. The end point of this tunnel is a Network Address Translator (NAT). This NAT bridges the hosts in Tarzan and the hosts that are not aware of Tarzan. Similarly, the NAT receives the response packets from the outside hosts and reroutes them back through this tunnel. In our protocol all the communication between the suppliers and the market is anonymized using a P2P network created by the suppliers.

4.2 Securing MAGNET

Based on the requirements mentioned before we now propose the following architecture to enhance the security of the MAGNET system. We outline our protocol into following phases: contracting, planning, bidding, auction close, and winner determination. Fig 2 illustrates the protocol in detail.

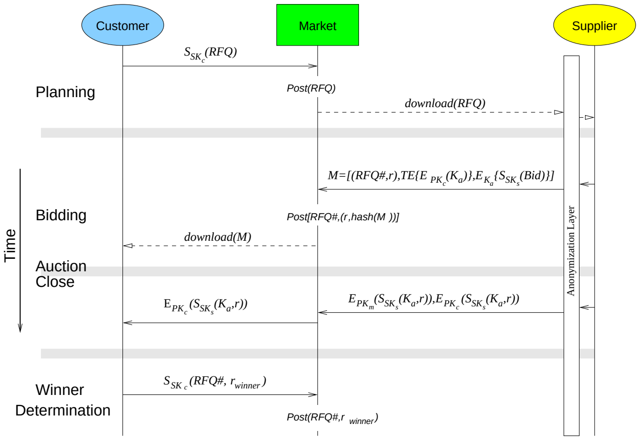

Figure 2: The Secure MAGNET system

4.2.1 *Planning:*

The customer sends a signed RFQ to the market for publishing.

$$Customer \xrightarrow{S_{SK_c}(RFQ)} Market$$

4.2.2 *Bidding:*

The supplier downloads the RFQ from the market. If interested, it generates a bid-message comprising of three parts:

1. General Information (GI): consists of the RFQ number and a sufficiently long random number, $(RFQ\#, r)$.

2. Auction-session key: a symmetric session key, K_a.

3. Bid data: comprised of price quoted by the supplier, the task list, the time-line for plan completion, *GI*, and supplier's public-certificate.

It then signs and encrypts the message and sends it to the market:

$$Market \xleftarrow{[(RFQ\#,r),TE\{E_{PK_c}(K_a)\},E_{K_a}\{S_{SK_s}(Bid)\}]} Supplier$$

For all the bid-messages received by the market, it posts *(RFQ#,(r,hash(M)))* on the market white-board, where M is the bid-message sent by a supplier to the market. The supplier can also check the white-board and verify that its bid was actually received and displayed by the market. The customer can then download the bids from the market. However, it cannot access the bid data unless it decrypts the time-release crypto. Since the exact timing of a time-lock puzzle is difficult to determine [11], the supplier agent would construct a puzzle that would take the customer longer than the auction deadline to solve.

4.2.3 *Auction close:*

Once the auction closes the suppliers would release K_a to the market in an encrypted form, alongwith the customer's copy:

$$Market \xleftarrow{E_{PK_m}\{SK_s(K_a,r)\},E_{PK_c}\{SK_s(K_a,r)\}} Supplier$$

The market would then pass on the customer's portion of the encrypted key:

$$Customer \xleftarrow{E_{PK_c}\{SK_s(K_a,r)\}} Market$$

This individual encryption is necessary so that no one except for the market and the customer can decrypt the bids.

4.2.4 *Winner determination:*

The customer agent uses various algorithms to determine the winner from the bids it has received [2, 3]. From the suppliers' certificates embedded in the bids, it can use statistical data to assist in the winner determination process. Once the winner has been determined, it would use the market's white-board to notify the suppliers about this.

$$Customer \xrightarrow{S_{SK_c}(RFQ\#,r_{winner})} Market$$

Once the market posts this result on the white-board any supplier can check to see if it is the winner. The customer agent can do a cross-verification by examining the white-board, in order to deter any wrong doing on market's behalf. The market can then carry out statistical aggregation by decrypting the bids using their respective auction-session keys.

5. ANALYSIS

5.1 Efficiency

The protocol that we have proposed tries to follow the original message exchange mechanism as closely as possible to avoid major redesign of the existing system. However, by utilizing a publish/subscribe system, the need for acknowledgment generation for each message has been eliminated. The suppliers and the customers can independently verify the data received by the other party, in an asynchronous manner, thus leading to better utilization of their resources. Using an anonymization layer can add some delay in message propagation, but the benefits are significantly higher.

Encryption and decryption of data is usually the most computationally intensive task in a security protocol. We have tried to minimize encryption of messages when possible. Thus, instead of encrypting the entire third section with time-release crypto mechanism, we introduced a second section, since encrypting and decrypting K_a is computationally cheaper because of the smaller size of the data. The task that could be most computationally intensive in our protocol is time-release decryption. However, this step is not needed as long as the customer waits for the auction to close and the supplier releases the auction-session key K_a.

Our motivation behind using a time-lock puzzle is more of a deterrent to prior opening of the bids and not to impose a computational penalty on the customer. However, in case the supplier fails to provide K_a (e.g. the supplier comes under a Denial of Service attack, or it refuses to the send the key on purpose), the customer can proceed with time-lock decryption. In such a case, if the customer faces a resource crunch, it can seek the market's help in decrypting it. The market has more computing power than the customer or the supplier agents since it has to provide the auction infrastructure. It can thus offer its resources for decryption to the customer. However, even after decrypting the time-lock puzzle it would not be able to get to K_a since it will be encrypted by customer's public key PK_c (for the same reason, anyone who intercepts the bid cannot get to the bid data). Thus the customer can safely shift the burden of time-lock decryption on the market, if required.

5.2 Security

Our security architecture overcomes the vulnerabilities in the existing MAGNET system. At the same time it tries to enhance the basic trusted third party model. We overcome the problem of secrecy of bids and the identity of the bidder by employing cryptographic techniques of anonymization and public-key encryption and decryption. By employing a time-release crypto mechanism, we ensure that the bids are not accessible by any agent until the close of auction.

We try to ensure the trusted third party model in MAGNET by including additional safeguards. By requiring that the market publish all the data which it received from the customers and the suppliers on the white-board, we enable cross verification of the data by the opposite party. The copies of the messages published by the market can be used to ensure non-repudiation in situations where the sender refuses to acknowledge an action.

Since the RFQs are publicly available for verification, it is difficult for the market to manipulate the closing time of the bids without being noticed by the customer. By making the market post the data related to all the bids received, we also ensure that it does not purposefully reject any messages.

As discussed before, prior opening of the bids is a form of collusion existing in the current auction systems. By not making the auction session-key (and the resulting bid data which it encloses) immediately available to the market, we limit the market-supplier collusion. There could still be a collusion between market and a supplier, but the market would be able to provide bid data only for the supplier agents who are already colluding with it. In absence of bid data for all the suppliers, a market-supplier collusion would not be useful. A collusion between all the suppliers and market would be similar to collusion between all the suppliers. This, as we discussed earlier, is beyond the scope of our current effort. The only visible information prior to decryption of the bid is the *RFQ#* and the random number *(r)* generated by the supplier. By enclosing the public certificate of the supplier with the remainder of the bid data, we ensure that the identities of the supplier agents are not known to the customer or the market before the end of the auction. However, once the auction is closed, the customer agent can use the information about the suppliers' identity to make its decision based on past statistical data.

6. RELATED WORK

The majority of the work in the field of auction security has been done on sealed bid auctions in which the auction outcome is made public. Most of these protocols require m auctioneers out of which at least n should be trustworthy (*threshold cryptography*). Franklin and Reiter proposed one of these earlier systems in which using a variation of the secret sharing scheme algorithm they try to reduce the trust on the auctioneer [7]. In their system the value of n is a third of m. The resulting system requires exchange of numerous messages making it highly inefficient. This system cannot be applied to MAGNET as there exists only a single instance of the market (auctioneer). SAM [9] is a system in which the trust is shifted from the auctioneer to a hardware implemented secure co-processor [14]. The basic idea is to replace the auctioneer by a combination of hardware and software which can be trusted by all the parties involved in the auction mechanism. Any tampering of the hardware results in its self-destruction. The system is similar to MAGNET in the sense that only a single auctioneer exists in the system. However, the market in MAGNET is not absolutely trusted as SAM is supposed to be. Protocols in which the auctioneer is completely eliminated have also been proposed [1]. The idea behind such protocols is to let the bidders decide the winner themselves by splitting and distributing all the bids amongst all the bidders. Using a secret sharing scheme, the bids can be assembled only if all the bidders are willing to do so. As long as a single bidder does not collude with the other bidders, individual bids cannot be determined. In case all the bidders collude, the auction mechanism becomes an "open cry" auction. Once the bids are reassembled the winner can be determined by finding the highest bid. This system cannot be applied to MAGNET as the winner determination process is more complex than simply determining the highest bid [2]. A system which utilizes temporarily secret bid commitment has also been proposed [13]. However, none of the systems described above meet all the security requirements faced by multi-agent marketplaces like MAGNET in the manner we have described.

7. CONCLUSION AND FUTURE WORK

In this paper we have presented ideas from recent work in security protocols for conducting secure electronic auctions and ideas for conducting agent-based electronic commerce. We have proposed a security architecture that would ensure the security of MAGNET and similar multi-agent marketplaces when used over public networks. Using various existing technologies, our protocol builds upon the trust model of the original marketplace and develops a system which has better methods of controlling fraud and deception.

We have begun implementing the proposed security model for use within MAGNET. Eventually we would like to release a secure version of the system to be used over the Internet. In the future we would also like to focus our work on establishing protocols for agent registration and payment collection in the execution phase once an auction has been closed. We would also like to include fault tolerance into the system so that the auction does not fail because of the failure of the entities involved.

8. REFERENCES

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

[2] J. Collins and M. Gini. Performance of winner determination search in combinatorial auctions with time constraints. *Submitted to the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2002.

[3] John Collins, Güleser Demir, and Maria Gini. Bidtree ordering in IDA* combinatorial auction winner-determination with side constraints. In J. Padget, Onn Shehory, David Parkes, Norman Sadeh, and William Walsh, editors, *Agent Mediated Electronic Commerce IV*, volume LNAI2531, pages 17–33. Springer-Verlag, 2002.

[4] John Collins, Scott Jamison, Maria Gini, and Bamshad Mobasher. Temporal strategies in a multi-agent contracting protocol. In *AAAI-97 Workshop on AI in Electronic Commerce*, July 1997.

[5] John Collins, Wolfgang Ketter, and Maria Gini. A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. *Int'l Journal of Electronic Commerce*, 7(1):35–57, 2002.

[6] John Collins, Ben Youngdahl, Scott Jamison, Bamshad Mobasher, and Maria Gini. A market architecture for multi-agent contracting. In *Proc. of the Second Int'l Conf. on Autonomous Agents*, pages 285–292, May 1998.

[7] M. Franklin and M. Reiter. The Design and Implementation of a Secure Auction Service. In *Proc. IEEE Symp. on Security and Privacy*, pages 2–14, Oakland, Ca, 1995. IEEE Computer Society Press.

[8] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, D.C., November 2002.

[9] Adrian Perrig, Sean Smith, Dawn Song, and J. Doug Tygar. SAM: A flexible and secure auction architecture using trusted hardware. In *First International Workshop on Internet Computing and E-Commerce (ICEC'01)*, pages 170–170, April 2001.

[10] Charles Phillips and Mary Meeker. The B2B internet report – Collaborative commerce. Morgan Stanley Dean Witter, April 2000.

[11] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical Report MIT/LCS/TR-684, MIT, 1996.

[12] Dawn Song and Jonathan Millen. Secure auctions in a publish/subscribe system. Available at http://www.csl.sri.com/users/millen/papers/dcca8.ps, 2000.

[13] Stuart G. Stubblebine and Paul F. Syverson. Fair on-line auctions without special trusted parties. *Lecture Notes in Computer Science*, 1648:230–240, 1999.

[14] Bennet Yee and Doug Tygar. Secure coprocessors in electronic commerce applications. In *Proceedings of The First USENIX Workshop on Electronic Commerce*, New York, New York, July 1995.