Security in Agent-Mediated Negotiation Frameworks

Richard Ssekibuule
Department of Computer Science
Faculty of Computing and IT,
Makerere University Kampala
rkayondo@cit.mak.ac.ug

Abstract

Several researchers in agent systems have proposed various settings for agent-mediated applications. As is the case with most software systems, there are security implications for any design decisions. This paper analyzes three agent-mediated negotiation frameworks and investigates the implications of their design to agent-mediated applications’ security. We present an overview of the unique threats faced by these negotiation frameworks, discuss implications of the threats and countermeasures so far suggested.

1 Introduction

In recent years, agent systems have been used in various real world applications such as distributed planning, scheduling, e-commerce and resource management. A key component to most agent-mediated applications in e-commerce, is the negotiation process through which entities participating in a transaction find a position of agreement in the event of a discrepancy. Negotiation has previously been defined [Filzermoser and Vetschera, 2008] as dynamic processes in which parties involved exchange offers, make concessions, or influence each other in order to reach an agreement. With the advancement of agent systems technologies, much research work has been done in integrating automated negotiation [Jennings et al., 2001; Bartolini et al., 2005] in agent-mediated applications. The minimum requirement for automated negotiation in agent systems is the ability to construct proposals and respond to requests.

Negotiation frameworks provide building blocks for facilitating automated negotiation processes. Functional requirements for negotiation frameworks have previously been defined by Mobach et al. [Mobach, 2007], and the design of a negotiation framework for AgentScape platform [Overeinder and Brazier, 2005] was based on these requirements. However, as is the case with several other complex software systems, several researchers have mainly focused on functional requirements and either ignored security completely or only considered it after advanced stages of functional designs. This situation is true for the negotiation framework suggested by [Mobach, 2007] and MAGNET by Collins et al.[Collins et al., 2002; Collins and Gini, 2006]. Some of these frameworks have considered implicit and explicit assumptions such as participants in the protocol being rational and not behaving maliciously towards one another. In a practical open distributed environment, such assumptions would be very strong due to high possibilities of a negotiation framework being compromised by internal and external attackers.

In the context of negotiation frameworks, we introduce a distinction between internal, external and agent attackers. “Internal attackers” refers to individuals and or entities that could be participating in the negotiation protocol. Such attackers could be people or software agents that are performing a bilateral trade or bidding in an auction. “External attackers” on the other hand refers to entities that are not participating in the negotiation protocol but have the ability to affect normal operations of the negotiation framework. Such attackers are assumed in the Dolev-Yao [Dolev and Yao, 1983] attacker model to have the ability of listening on the communication channel, intercept and analyze messages being exchanged between communicating parties. “Agent attackers” have stronger abilities than the internal and external attackers to perform malicious activities on the negotiation framework. A malicious platform which has the ability to terminate an agent or destroy its data is an example of agent attacker.

In the design of negotiation frameworks, various application settings have been proposed but no comparative discussion on the effect of these application organizations to the security of the negotiation frameworks has been done. We consider three negotiation frameworks for this comparative study.

2 Design of Agent-Mediated Negotiation Frameworks

As previously indicated in the introduction section, negotiation frameworks facilitate automated negotiation processes. This is particularly important in order to achieve automation in agent-mediated negotiation. In this section we present details of negotiation building blocks, the activities involved, stakeholders and
Negotiation Protocol: The protocol defines a set of rules and procedures that have to be adhered to by parties participating in a negotiation. In e-commerce, negotiation protocols ensure that participants adhere to the trading rules. In the example of auction markets, a negotiation protocol will impose rules on participants for the type (Dutch, English or Vickrey) of auction being carried out.

Negotiation Objects: These are the range of issues over which participants in negotiation can make choices. Such issues may include price, quality, model of item and service agreements.

Agent Decision Making Model: The agent decision making model works within the provisions of the negotiation protocol and negotiation objects available. The quality of the agent’s decision making model determines the level of success that the agent will achieve.

In the subsections that follow, three example agent-mediated negotiation frameworks are presented. These negotiation frameworks were chosen because of their differences in architectural design. In order to have easy references in the article, new labels (names) are assigned to the two negotiation frameworks without short names. These three agent frameworks are used to highlight differences in threats that may occur in agent-mediated negotiation frameworks when certain design decisions are made.

2.1 TrinAge

This negotiation framework relies on three agents (Controller Agent (CA), Worker Agent (WA) and Itinerary Register Agent (IRA)) to carry out the negotiation process on behalf of the consumer: The negotiation process involves a client (represented by three agents) interacting with shortlisted suppliers on certain parameters (negotiation objects) until they reach an agreement or abort the negotiation. The framework depends on a trusted host to provide a trusted computing base for performing business critical services such as setting negotiation parameters, evaluation of offers and decision making [Al-Jaljouli and Abawajy, 2007].

- The Controller Agent (CA) stores critical data such as the list of offers, expiry time of the bid, scoring and decision functions.
- The Worker Agent (WA) stores non-critical data and tactic functions which form the agent’s decision making model.
- The Itinerary Register Agent (IRA) stores addresses of the visited suppliers and the time (t) at which the agent got executed at the service provider’s host.

TrinAge assumes a scenario in which agents (WA & IRA) representing a customer move through public networks and to possibly untrusted service provider hosts in search of offers. The framework aims to achieve confidentiality of negotiation strategies and decision making model by separating the handling of critical and non-critical data into two agents (CA and WA respectively). Figure 1 below, presents an overview of the TrinAge architecture. It is worthy noting that the supplier (service provider) entity is not an agent.

Figure 1: TrinAge Architecture

2.2 MoVir

Mobach et al. [Mobach, 2007], designed a negotiation framework in which consumers negotiate with service providers through mediators. In this framework, consumers are presented with services aggregated in virtual organizations. The creation and management of virtual organizations is performed outside the negotiation framework. Consumers can negotiate with several mediators and mediators simultaneous negotiate with several service providers. Service provider details are hidden from the customers by the mediators that provide an aggregated point of contact. The application framework is very dynamic, involving service providers leaving and joining virtual organizations at any time. The framework consists of a Consumer Agent (CA), the Service Provider Agent (SPA) and Mediator Agent (MA) whose roles are briefly explained below:

- The Consumer Agent (CA) represents the human buyer in the negotiation process. Consumer agents initiate the negotiation process by looking for services. Consumer Agents locate mediators and the services they offer through some external services such as directory services [Moreau, 2001]. The CA performs multi-attribute negotiations with one or more Mediator Agents (MA). Customer Agents may have to migrate to Mediator Agent platforms in order to accomplish their tasks.
- The Service Provider Agent (SPA) is responsible for providing goods and services that are sought
after by consumer agents. Depending on the service type, the SPA may be required to migrate to the Mediator Agent (MA) platform to provide the requested service.

- The Mediator Agent (MA) represents a virtual organization of service provider agents that are governed by specified policies. Each virtual organization is administered by service policies that are different from other virtual organizations.

MoVir assumes an environment in which the customer agent and agents in the virtual organization (mediator and service provider agents) execute from trusted platforms. Figure 2 below presents an overview of the MoVir negotiation framework.

![MoVir Negotiation Framework](image)

**Figure 2: MoVir Negotiation Framework**

### 2.3 MAGNET

The Multi-Agent NEgotiation Testbed (MAGNET) is an architecture that was developed to provide support for agent interactions in trade negotiations. MAGNET supports automated contracting, negotiation and monitors execution of contracts in business transactions [Jaiswal et al., 2004; Collins and Gini, 2006]. MAGNET provides an automated marketplace through which software agents represent their owners in an auction. In order for the consumer agent to fulfill a set of tasks, it generates a plan from the tasks and solicits for help from supplier agents in the market. Essentially, this framework comprises of two agent roles: the customer and supplier roles.

Figure 3 below presents an overview of the MAGNET architecture in which a trusted third party (market) mediates all communication between customers and suppliers participating in an auction. The customer agent sends out Requests For Quotes (RFQ) to suppliers in the market and a set of suppliers with the requested services would reply with an indication of prices and any other constraints such as time. After the customer agent has received the bids, it evaluates them based on predefined parameters such as price and time and then selects the most optimal bids. After the bids have been selected, successful suppliers are notified and the Execution Manager is called to oversee completion of the negotiation process with the shortlisted suppliers. The Planner is called to renegotiate offers which are then sent to the Bid Manager for re-assignment to the Execution Manager. The execution manager, planner and bid manager are components of the customer agent.

In Table 1 below, we present a summary table comparing assets that can be found in the three negotiation frameworks. The items presented are a representation of underlying structural details in the negotiation frameworks.

### 3 Stakeholders and Assets in Negotiation Frameworks

Examining section 2 and subsequent subsections (2.1, 2.2 and 2.3) reveals that all the three negotiation frameworks (TrinAge, MoVir, and MAGNET) are comprised of a customer and service provider at a minimum. In some circumstances, such as MoVir and MAGNET, a moderator or trusted third party is used to oversee and facilitate activities between customers and suppliers. In addition to the customers, suppliers and moderators, agent-mediated negotiation frameworks consist of other assets such as negotiation protocols, negotiation objects or parameters and decision making functions that need to be protected. Typically, customer agents will carry negotiation parameters and decision making functions which are used during execution of the negotiation protocol. Customer agents are responsible for receiving bids from suppliers. Adversaries in the environment will attempt to compromise these assets either for their own profit or to fail goals of legitimate participants.

The following common elements have been identified in the three negotiation frameworks: (a) Agents that are participating in the negotiation. (b) Offers or sometimes referred to as services. (c) Bids submitted in response to a request for offer or quotation. (d) Negotiation objects or parameters. (e) User and market predefined constraints such as delivery time, bid closing time and expiry time of offer. (f) Shortlist of suppliers to be considered by the customer. (g) Contracts between suppliers, market/moderators and customers.

### 4 Mobility Comparisons

Mobility features are critical to security of agent systems. This is because agents may migrate to malicious platforms. Agents executing on maliciously platforms face greater risks of having security properties of their code or data being compromised. Due to design differences in negotiation frameworks, some agents are expected to migrate to other platforms in order to
Table 1: Comparison of Assets in the Negotiation Frameworks

<table>
<thead>
<tr>
<th>Framework</th>
<th>Entities Involved</th>
<th>Agent?</th>
<th>Mobile?</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrinAge</td>
<td>Customer</td>
<td>Controller</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Itinerary Register</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Worker</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Supplier</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MoVir</td>
<td>Customer</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Mediator</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Service Provider</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MAGNET</td>
<td>Customer</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Market</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Supplier</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2: Comparison of Entities and Mobility in Negotiation Frameworks

accomplish their tasks while in some frameworks migration is not expected. Moreover, in the case of TrinAge, the service provider is non-agent. Table 2 below presents comparisons of entities in the various negotiation frameworks which are agents, non-agent, mobile and non-mobile components.

5 Attack Model and Trusted Computing Base

Figure 4 below provides a high level representation of possible interactions between agents in negotiation frameworks. It illustrates a scenario in which Agents A and B might be involved in a bilateral negotiation with a supplier agent. Agents A and B are legitimate participants in the transaction with a supplier, but agent B may wish to cheat agent A using the knowledge he or she possess about the negotiation protocol. In the listing below, we describe the three forms of attacks identified in negotiation frameworks.

i. Malicious Agents: This type of attacker has privileges of legitimate participation in the negotiation. The attacker has the ability to study the negotiation protocol and maliciously use this information against other participants in the negotiation. A customer agents may collude in the market to unfairly out compete other agents.

ii. Malicious Platforms: This is a powerful adversary that can compromise confidentiality, integrity and availability of the agent and its data. These attackers can have the ability to intercept
intraplatform messages, to destroy the agent and its data structures.

iii. Dolev-Yao Attacker: In the Dolev-Yao threat model [Dolev and Yao, 1983], an attacker is assumed to have the ability to listen on the communication channel and can intercept messages being exchanged between two parties. In our scenario of negotiation frameworks, such attackers can intercept intraplatform and inter-platform message exchanges.

Unlike TrinAge, MoVir and MAGNET assume a trusted platform for execution of all participating agents. TrinAge assumes a scenario in which customer agents may visit untrusted platforms and thus provides protection mechanism for survivability of the customer agent by splitting functionality between three agents. However, a malicious platform could still destroy an agent that migrates to a hostile platform.

Table 3 below presents a summary of attacks that are possible with the three negotiation frameworks based on their design (see section 2) and stakeholders (see section 3) involved. MoVir and MAGNET assume a trusted platform for agent execution, hence eliminating possibilities of an attacker agent. TrinAge caters for situations in which the platforms are not trusted. The Worker and Itinerary Register Agents are tasked with the risk of collecting information from untrusted service provider platforms and returning to a trusted platform to perform business critical functions.

6 Attacker Goals and Attack Vectors

This section presents various threats that can be faced by the negotiation frameworks and the means by which adversaries would achieve their goals. Confidentiality, Integrity, Availability and Non-repudiation (CIAN) are well studied security properties for digital systems; this section presents an overview of attacks that could be used to compromise these security properties in reference to the assets and stakeholders involved. However, because of threats which required more details than can be achieved with CIAN, additional requirements are presented in the subsections which follow.

6.1 Confidentiality Attacks

In all the negotiation frameworks considered, message exchanges are involved and participants would desire to keep their contents confidential either for privacy reasons or to enable them compete fairly with other participants. Such messages include requests for offers, requests for quotations, negotiation parameters, user predefined constraints, bid messages and contracts with suppliers and market moderators. Additionally, an attacker might be interested in maliciously disclosing identities of agents that participated in a particular transaction. Adversaries could also be interested in knowing about details of the bids before the auction closing time so that they make better bids than their competitors.

6.2 Attacks against Integrity

Integrity threats are targeted at changing contents of messages, agent code and other objects such as negotiation parameters, closing time of bids and offers in order to suite interests of an adversary. An attacker could for example change contents of the list of shortlisted suppliers in order to prevent the customer from considering certain suppliers in the final negotiation phases. In case the bid closing time or expirt time of an offer is changed by an adversary, participants would not be able to rightly accomplish their goals. A more complicate integrity problem also exists in frameworks which support mobility. An adversary could for example launch a man-in-the-middle attack which could be used to alter the agent’s code. If the code checksums are not protected, the attacker could even recompute the checksums and insert a new version of a compromised agent.

6.3 Attacks on Availability

Denial of service attacks are a threat to proper functionality of all services involved in the agent-mediated negotiation framework. Survivability of an agent is at a higher risk when the negotiation framework supports mobility. Unlike TrinAge, MoVir and MAGNET assume a trusted platform for execution of all participating agents. TrinAge assumes a scenario in which customer agents may visit untrusted platforms and thus provides protection mechanism for survivability of the customer agent by splitting functionality between three agents. As previously indicated in Table 3, an Agent attacker could still destroy an agent that migrates to untrusted platforms. Moreover, other assets such as offers shortlisted from suppliers could be removed from the shortlist by a competing supplier. Messages exchanged during the negotiation process could be hijacked by an adversary so that they do not reach intended participants.

Publication of services in the publish/subscribe system such as the one used in MAGNET and MoVir could be done in a such a way that some participants do not get the information timely. Such a scenario could lead to unfavorable competition amongst participants in-case time constraints are involved in a bid. Some adversaries could also publish bogus bids into the publish/subscribe system or even replay bids from other suppliers to damage their reputation or that of the entire market. In circumstances where negotiation frameworks rely on directory services to discover markets or suppliers (e.g. MoVir), a compromise to a directory service would imply that the suppliers will not be discovered. This kind of attack is possible with MoVir where a mediator agent aggregates services from suppliers in a virtual organization and assumes that all parties in the negotiation framework are trusted. Attackers could also use more resources than they are authorized on service provider platforms. Such actions would lead to denial of services to other agents accessing related services.
6.4 Non-Repudiation Attacks
Non-repudiation means that there is proof for actions that were performed by participants in a negotiation. A customer should not be able to deny having submitted a request for an offer from suppliers. The negotiation framework should also ensure that the supplier cannot repudiate the responses they made to their customers. In MoVir, a mediator agent could deny having received service subscriptions from some suppliers. In order to achieve non-repudiation, some form of identification is required, the negotiation framework should additionally provide protection for identification information so that adversaries do not use it to violate confidentiality requirements.

6.5 Collusion Attacks
Formation of coalitions in negotiations can be advantageous to both suppliers and customers. Customers get a chance to bargain for lower prices due to expressions of higher demand (with bulk purchases) for products and services. On the other hand suppliers get an opportunity for bulk sales. However, unsupervised coalitions can also be used by colluding buyers to unfairly out-compete other participants. In situations of bidding, suppliers may not receive competitive prices if customers collude to purchase services at lower prices. All the three negotiation frameworks studied are vulnerable to collusion attacks.

7 Security Mechanisms in Negotiation Frameworks
This section presents security mechanisms that have so far been suggested in the negotiation frameworks to mitigate some of the identified threats. The solutions presented here are categorized based on the threats that they mitigate.

7.1 Confidentiality Countermeasures
Faced with threats of malicious disclosure of agent identities, anonymity can be used to prevent this threat from happening by hiding identities of participants. However, almost all transactions involved in agent-mediated negotiation would require some form of accountability and/or authentication. It would be inappropriate to have true anonymity yet agents may have to be accountable for some of their actions.

We instead recommend pseudonymity for such environments. Which provides pseudo names to agents so that their actions cannot be authenticated and recorded. Pseudonymity is also relevant for frameworks that reply on agent reputations to build trust models. Reputations cannot be built in environments that do not have any reference point to past activities of agent identities. An example of security schemes using pseudonymity (referred to as virtual identities) have previously been suggest by [Farkas et al., 2002] and [Ziegler et al., 2006]. Practical implementations of pseudonymity have been achieved in AgentScape [Warnier and Brazier, 2009] platform by use of agent handles as pseudonyms to support authentication and anonymity in payment schemes.

Other issues of confidentiality also arise for example in auctions whereby information about the bids is not supposed to be revealed to participants until the end of the auction. Techniques such as time-lock puzzles and timed-release cryptography [Jaiswal et al., 2004; Rivest et al., 1996] have been suggested in systems like MAGNET to prevent such threats. Lastly, encryption of messages, use of signed digital certificates for communication between negotiating parties and mutual authentication would also prevent eavesdropping, spoofing and man-in-the-middle attacks in negotiation frameworks. These techniques can be implemented using the Transport Layer Security (TLS) protocol [Dierks and Rescorla, 2008]. However, challenges still exist in regard with threats from internal attackers that may also have valid certificates with the Certificate Authority and can establish legitimate connections between targeted communicating parties.
7.2 Integrity Countermeasures

As indicated in subsection 6.2 agent-mediated negotiation frameworks face integrity threats against their own code and the information they carry. Countermeasures to integrity threats can be classified into detection and prevention categories. Majority of the techniques discussed here are detection techniques for tampering either on agent data or code. TLS protocol implementations can be used to guarantee integrity of messages and mobile agent code against an external attacker but will not protect the negotiation framework against an internal attacker. Similar to attacks on messages, man-in-the-middle attacks can be launched by an internal attacker on mobile-agent code. Mobile agent code could be changed and new versions of agent code inserted by adversaries during transactions. Such threats can be detected by signing agent code and data objects. Ideally agent code should be signed by the agent creator/developer and the data objects digitally signed by the agent owner. Security mechanisms involving signing of code and objects would require a Public-Key Infrastructure (PKI) for retrieve decryption keys. A PKI is also important for mutual authentication and providing secure communication channels using TLS between communicating parties.

A distributed trust technique was proposed by [Warnier et al., 2008] to maintain integrity of migration paths so that in case of deletion or tampering with agents' data or code, the malicious platforms can be held accountable. The other two known mechanisms for recording and protecting migration paths involve (a) using a centralized trusted third party to authorize and keep track of migration paths and (b) using signature chaining without a trusted third parties [Saxena and Soh, 2005]. The distributed trust technique provides better mechanisms for protecting migration path information. It avoids the overheads of a trusted third party and mitigates problems involved with storage of signatures of migration paths inside the agents themselves by using signed waivers of migrations path information from previously visited hosts.

All negotiation frameworks reviewed in section 2 implicitly assume trust for the agent. However, this assumption cannot always hold since agents may come from malicious customers or they (agents) might have been compromised before arrival at the execution platforms. Sand-boxing or jailing [Balogh et al., 2007] provides one of the well tested techniques for protection of platforms against execution of untrusted code. Model Carrying Code (MCC) [Sekar et al., 2003] is another approach which provides a practical approach to monitoring of untrusted mobile code execution on agent platforms. Compared with Proof Carrying Code (PCC) [Necula, 1997], MCC overcomes limitations of verifying binary code after application deployment [Sekar et al., 2003]. The mobile-agent is expected to present a high-level model of its security-related behavior to the agent platform to be verified for conformance with the security policies of the platform.

7.3 Availability Countermeasures

As a first step for protecting agent-mediated systems against availability threats, detection mechanisms based on the techniques discussed in subsections 7.1 and 7.2 should be put in place. Security mechanisms such as recordig of migration path using distributed trust can be used for accountability in case agent gets changed or destroyed.

It is also critical to manage privileges of agents on agent platforms to prevent denial of service attacks. Schemes such as Role Base Access Control (RBAC) [Quillinan et al., 2008] have been implemented in agent platforms such as AgentScape to manage roles and access to system resources. However, this mechanism also introduces challenges to provision of anonymity services to agent-mediated negotiation.

Threats of agents being destroyed when they migrate to malicious platforms have not been adequately dealt with. Most security mechanisms provide for detection of agent failure and tampering, but no solid solution has been presented for graceful failure or protection of data owned by the destroyed agent. A faulty tolerance technique based on replication of an agent had previous been presented by [Tomas and Tschudin Christian, 1998]. However, complete agent replication schemes impose computational overheads on agent platforms.

The last problem we consider in this category of threats is annoyance attacks in which an attacker may make bogus submissions to the publish/subscribe system. Agent platforms such as AgentScape [Quillinan et al., 2008], JADE [Gunupudi and Tate, 2006], SeMoA [Roth and Jalali, 2001] and Cougaar [Helsinger et al., 2004] have all implemented identity management schemes to facilitate authentication and authorization requirements. However, only AgentScape has implemented partial anonymity services which can be used for authentication and authorization. We refer to the anonymity services provided by AgentScape as partial because they use agent handles (which is a hash of the global agent identifier) which can be traced back to the true identity of the agent by an authorized administrator. It is worthy noting that MoVir framework was implemented using AgentScape platform and thus utilizes all security services in AgentScape. The question of whether an agent should be allowed to submit offers or not to a publish/subscribe system or to a mediator agent as in the case with MoVir is largely a trust issue. The agent systems research community [Fullam et al., 2005] has largely considered reputation models for answers to trust relations.

7.4 Repudiation Countermeasures

Psuedonymity can be used to maintain partial anonymity and accountability for agent actions when participating in negotiations. Use of reputational models in such environments works as good incentive to correct behavior to participating agents. Digital signatures can be used to provide proof that a certain party sent a message. A message sender cannot claim
that they did not send a message that they signed unless their private key was stolen. Identity Management Systems for agents in the execution environment have been implemented for example in AgentScape to prevent an agent owner from repudiating ownership of an agent. Agents are identified by the middleware using a global unique identifier (GUID). The GUID is private to the middleware and cannot be used by elements in the agent environment to send or receive messages from the agent. Instead a hash of the GUID concatenated with a counter is generated and published to a directory service for public access. All agents in AgentScape are signed by the agent owner to provide non-repudiable ownership. Unlike MoVir whose security mechanisms are based on AgentScape middleware, TrinAge and MAGNET frameworks do not have security mechanisms for providing both anonymity and accountability services.

7.5 Collusion Countermeasures

A solution to collusion attacks based on public verifiability of winner determination data in case of a dispute was presented by Shi et al. [Shi et al., 2008]. Participants with secret keys to winner determination data would be able to retrieve the decision making data. This solution is inadequate and cannot address the problems presented in subsection 6.5 since collusion trails cannot be found in decision making data.

8 Discussions and Conclusions

This article has taken a detailed study of assets and stakeholders involved in the negotiation frameworks and investigated possible risks that would be experienced in the event of an attack. An attack model that is based on internal, Dolev-Yao and Agent attacks is discussed and applied to the three negotiation frameworks. Standard cryptographic techniques are available to prevent most Dolev-Yao based attacks, but the design choices for implemented security protocols may introduce vulnerabilities. This scenario can be found in the security protocol presented by Jaiswal et al. [Jaiswal et al., 2004] to mitigate weakness in MAGNET, but did not cater for replay attacks as noted by Shi et al. [Shi et al., 2008]. TrinAge makes effort to protect components of the customer agent against malicious platforms, but does not completely achieve this because the information collected by the worker and itinerary register agents (which visit hostile platforms) could still be destroyed with these two agents. TrinAge further provides mechanisms for detecting when an offer has been removed from the collection of offers, but does not provide prevention mechanisms against this attack before offers are submitted to the controller agent. The publish/subscribe system proposed in MAGNET provided a more secure centralized control for market transactions than the mediator agent in MoVir and the mobility functionality in TrinAge. In table 4 below, we present a summary table mapping security requirements in negotiation framework to various security mechanisms. As much as this article presents security mechanisms that can be used/deployed to mitigate threats, it does not address weaknesses in these security mechanisms. Security problems that exist with reputational models, publish subscribe systems, distributed trust and security protocols are not addressed in this paper and remain areas of future research.

Acknowledgments

Many thanks to Erik Poll and Jose Ghislain Quenum for the tireless reviews and security insights.

References


Table 4: Security Requirements and Mechanisms

<table>
<thead>
<tr>
<th>Requirements/Security Mechanisms</th>
<th>Requirements/Security Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality (Identities and Requests/Bids)</td>
<td>Encryption, Pseudonymity, Timed-release cryptography</td>
</tr>
<tr>
<td>Integrity (Messages and Agent Code)</td>
<td>Digital signatures and certificates, migration paths</td>
</tr>
<tr>
<td>Availability and Fairness (Agent and Messages)</td>
<td>Role-based access control (principle of least privilege), Sandboxing techniques, migration paths, Reputation models</td>
</tr>
<tr>
<td>Trusted Service Discovery</td>
<td>Authentication and authorization techniques, Reputation models</td>
</tr>
<tr>
<td>Non-repudiation</td>
<td>Authentication, digital signatures and certificates</td>
</tr>
<tr>
<td>Collusion Prevention</td>
<td>Reputation models, “Public” verification of information</td>
</tr>
</tbody>
</table>


[Warnier and Brazier, 2009] Martijn Warnier and Frances Brazier. Anonymity Services for Multi-
