Specifying Standard Security Mechanisms in Multi-Agent Systems

Stefan Poslad[1], Patricia Charlton[2] and Monique Calisti[3]

[1]Department of Electronic Engineering, Queen Mary, University of London
stefan.poslad@elec.qmul.ac.uk
[2] Motorola Labs, Espace technologique St Aubin, 91193 Gif-sur-Yvette Cedex - France
charlton@crm.mot.com
[3]Whitestein Technologies AG, Gottardhstrasse 50, CH-8002 Zürich
mca@whitestein.com

Abstract. Distributed multi-agent systems propose new infrastructure solutions to support the interoperability of electronic services. Security is a central issue for such infrastructures and is compounded by their intrinsic openness, heterogeneity and because of the autonomous and potentially self-interested nature of the agents therein. This article reviews the work that the FIPA agent standards body has undertaken to specify security in multi-agent systems. This enables a discussion about the main issues that developers have to face at different levels (i.e., intra-platform, inter-platform and application level) when developing agent-based security solutions in various domains.

1 Introduction

In the same spirit that the Internet developed for access to information, there is a vision to sustain open service environments with an e-business model that supports dynamic services, automated interaction, rich information exchange and tailored solutions. The research and development of Multi-Agent Systems (MASs) has often targeted the provision of infrastructures for e-business solutions within open service environments. Multi-agent systems represent virtual societies where software entities (agents) acting on behalf of their owners or controllers (people or organizations) can meet and interact for various reasons (e.g., exchanging goods, combining services, etc.) and in various ways (e.g., creating virtual organizations, participating to auctions, etc.). When deployed in an open environment such as the on-line business world, multi-agent systems face particularly challenging trust and security issues at various levels.

However, a major problem is that only very specific areas within the Internet space offer advanced security solutions to protect both service providers and consumers against malicious attacks. Furthermore, these “secure islands” are typically centralized closed systems that heavily rely on human supervision and control. Internet users are becoming increasingly aware of security problems such as experiencing fraudulent transactions, even without having used the particular on-line service for

which that transaction occurred. As electronic information and services are handled more automatically on behalf of the user, the user no longer knows how and what data is secure in the electronic exchanges.

As agent technology and the support infrastructure advances, they offer the potential to help support the enhanced security requirements of more open service environments. However, the problem of security and in particular agent security is a very multi-faceted issue that in the real world involves trade-offs, unseen variables, and imperfect implementations. A good security design will define a system architecture supporting the relationships between prevention, detection and reaction [1]. However, highly distributed open service systems, such as MASs currently have no coherent theory, architecture design and implementations to use even classic Internet security in a standard way. If "openness" is to be key, as it brings many advantages to the deployment of services and information, then security that covers the many needs of the environment, services and applications, requires some basic security standards.

1.1 Trust, security and privacy

Current research has demonstrated that we bring our social model to the world when we interact with various inanimate objects from the toaster to the computer [2]. For example, our very social and cultural approach to evaluating a first meeting of a service can be strongly influenced by someone's recommendation if we have attributed a high-level of creditability of knowledge to a person concerning that particular service. Hence, the very success or failure of a service in the physical world could be based on someone's recommendation. The multifaceted nature of trustworthiness requires support for the generic concepts of trust, security, and privacy [3]:

Trust is a social concept for evaluating risk, which is often situated in a cultural environmental and is driven by a community's need for cooperation through communication and interactions for the perceived survival of that community.

Security is a set of physical realizations that reduces the risk of potential hazards when interacting with the environment. Social trust does not necessarily need to have security. However, security can provide the fundamental building blocks for supporting concepts of trust. The mainstream computer network community also uses a concept of trust associated with a network of trusted third parties that are introduced in order to approve unforgeable bindings between names and objects such as public encryption keys, roles and access control lists. It is assumed that this belief in these bindings is complete by all parties. We refer to this specific concept trust as encryption trust.

Privacy provides both a conceptual and physical space for the social protection of high-valued items such as knowledge, information, objects, services, that a person or community places a high-value on and that these items are respected as such. Often privacy utilizes both the concepts of security and trust.

The remainder of this paper is structured as follows: In section 2 the Security requirements are generated from a set of use-cases, section 3 discusses some main issues in standardizing agent security. The security related FIPA specifications are reviewed and the use of the FIPA specifications for secure MAS systems are ana-

lyzed. Finally, a discussion about future directions for standardizing MAS security concludes the paper.

2. Requirements and Use-Cases

E-business Open Service Spaces (OSSs) are characterized by: heterogeneous service components from multiple providers; dynamic service mergers where multiple autonomous domains may become interlinked and loose some of their autonomy; dynamic service demergers, information that is distributed across insecure environments and richer interactive information exchange that can span multiple domains.

In order to illustrate some of the pertinent issues and to generate requirements for MAS security within OSSs, some security related scenarios (see Table 1) have been modelled by the FIPA Security workgroup as part of a white paper [4].
For example, in the privacy and personalization scenario, agent interaction in a medical environment is modelled. A personal agent A manages a person's preferences and characteristics such as tolerance to drugs, gender etc. for a human principal. A doctor service agent B provides the medical help and is able to access these preferences and characteristics in order to slant an instance of a service invocation to that agent, i.e., to treat a patient's medical condition. Other hospital agent services C, D may be used by agent B to carry-out its service and finally other personal agents E and F may also talk with agent A to find out about information about C's service.

Table 1. Some application scenarios and their main security issues

Scenario	Security issues
Publisher/directory	Authentication, authorization, DoS,
Courier/broker scenario	Message privacy, integrity, authentication, non-repudiation
Task Allocation scenario	Non-repudiation, contract integrity, message privacy
Multi services domains scenario	Propagation of authentication, authority, trust across multiple domains
Personalization and privacy service scenario	Privacy & integrity of user preferences, privacy & integrity of service capabilities, authentication of owner, action, policy integrity & privacy, trust
Mobile agent application scenario	Agent integrity, message integrity

The following security problems can occur within this specific scenario:
The service agent B may divulge private information (a user's personal preferences) to other service agents C and D against the wishes of the user agent A;
The user agent A may reveal its favourable service offer to other personal agents E and F against the wishes of the service agent B;

The identity of A's human-owner or principal may be modified so that A is associated with different characteristics and so receives an ill-matched treatment plan;
The personal agent policy for revealing his or her preferences and characteristics to a specific agent such as a doctor agent may become compromised, e.g., the new policy is now that the user agent can reveal information to any other agent;
Another agent, who is not qualified to offer a doctor service, may masquerade as an instance of a doctor service type;
A may trust a particular doctor B to treat A, but B may become replaced by another instance of the doctor agent.

These simple examples are just a subset of even more complex situations that may occur in a number or various real applications and environments. Basically, the threats in the digital world mirror the threats in the physical world. However, whereas we have systems such as trust (albeit not perfect) in the physical world in order to provide supplementary types of protection necessary for the type of service or situation, we lack such mechanisms in the open digital society.

3. MAS Security Models

The Foundation for Intelligent Physical Agents or FIPA, a forum of international companies with a strong focus in the telecommunication industry, was formed in 1996 to promote the uptake of software agents in businesses at large [5]. It focuses on supporting MAS interoperability and has produced a number of specifications in this area. The first FIPA specifications were released in 1997. In 1998, FIPA first became active in specifying agent security [6]. This initial specification has since been made obsolete, but it has provided some useful hooks to model security within a FIPA agent platform. In the following sections, we first present a general MAS security model and then review some different approaches to providing MAS security.

3.1 Architectural MAS Security Elements and General security requirements

The FIPA abstract architecture specification [7] covers some of the general properties for security, but it stopped short of proposing one or more (abstract) functional architectural elements for security such as secure channels or authentication services. The security concepts in the abstract architecture are summarized here: The central requirements for security are:

Authentication: the ability to determine the identity of the various entities in the system.
Authorisation: based on the identity of an entity, determine what access policies apply to the entity.
Integrity: the ability to determine whether a piece of software, a message, or other data has been modified since being dispatched by its originating source.
Privacy. The ability to ensure that only designated identities can examine software, a message or other data. To all others, the information is obscured.

In more concrete terms, FIPA has specified security for specific services such as the message transport service [8] and agent management [9], [6]. The security models in these specifications never matured sufficiently and as a result, in practice, security is added to FIPA MAS systems in a variety of non-FIPA specified, proprietary, ways.

3.2 General MAS Security Asset Model

We can view security generally as a set of safeguards that help to protect the assets - the items of value in a system. These safeguards protect the system assets against threats that seek to disrupt the operation, integrity and confidentiality of these assets. Different security views or profiles specify a set of one or more safeguards to protect these assets against threats.

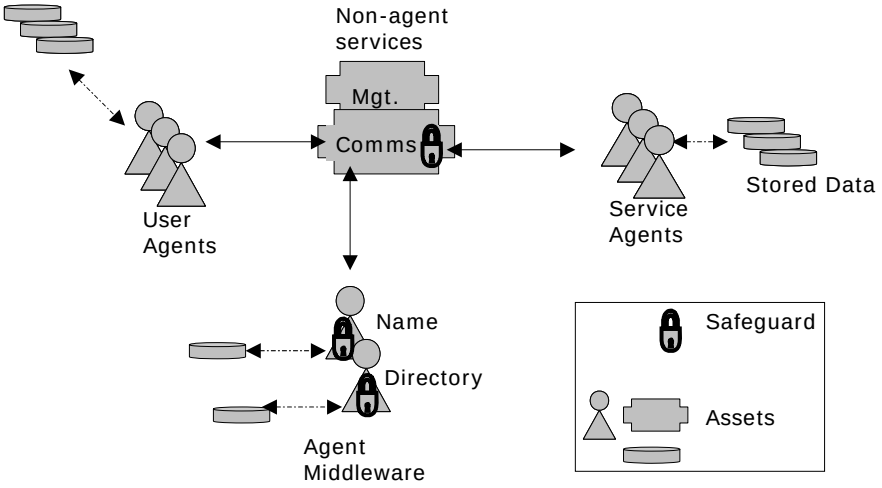

Fig. 1. Simple Safeguards (depicted as padlocks in the diagram) for FIPA MAS Assets (shaded elements). MTS represents a communication sub-service and Mgt refers to an agent management sub-service.

In the simple FIPA Asset Security Model, we can identify the assets as:
Agents: these are user agents, service agents and middleware or middle agents such as name services and directory services. Although the FIPA agent management specification specifies name and directory services as agent-services, the FIPA abstract architecture defines the name and directory service more generically –they can be represented as NA-services or agent services.
NA-services or Non-Agent Services: there are certain entities in the system that for a variety of reasons such as performance and because of existing practices, are not represented as agents. NA services include the Message transport Service, the agent management service that loads, starts and stops agents and data storage services.

The most common MAS safeguards seek to guard the communication NA service and the middleware services such as the naming and directory services (see Table 2). The FIPA name service defined in the FIPA Agent Management Specification can actually support agent management functions but in practice many FIPA MAS systems support agent management as a NA service.

Table 2. A basic security profile that links threats, safeguards to three of the core MAS assets: communication, name and directory service.

MAS Assets	Threats	Safeguards
Communication Service	-Corruption of transmitted data -Eavesdropping	-Signed or hashed messages -Encryption of transmitted data
Name Service	-Faking identify in a message exchange or service request	-Use of signed credentials from trusted parties.
Directory Service	-DoS (Denial of Service) -Unauthorised write access	-Access control to directory until agent becomes trustworthy -Verification of requesting agent & use of authorisation lists

We can think of Table 2 as defining a very basic security profile that links threats, safeguards to three of the core MAS assets: communication, name service and the directory service. We next examine some of these MAS assets and their safeguards in more detail. We first examine the safeguards that FIPA has specified then we examine how additional, non-FIPA specified MAS safeguards have been used by the agent community.

3.3 Message Communication

MAS systems are fundamentally message-based, therefore threats to the communication such as corruption of transmitted data and eavesdropping need to be guarded against. Safeguards for communication can be divided into two types: asynchronous message safeguards and synchronous message safeguards. The early safeguard designs for FIPA MAS communication focussed on autonomous agents and asynchronous communication.

A traditional safeguard for asynchronous communication is to provide a secure envelope for each message sent, on a per message basis. For example, the FIPA Message Transport Service (MTS) specification [8] specifies an optional tag in the message transport envelope called “encrypted” - this defines how an ACL message can be encrypted for exchange between two agents. The use of this encryption follows the IETF RFC822 model [10]. The value of the envelope encrypted field is optional. The majority of the MTSs implemented in practice, in MASs based on open-source software implementations of the FIPA specifications, such as FIPA-OS, JADE and ZEUS, do not support this optional encrypted envelope field specification. If how-

ever, the message envelope encrypted field is set, it indicates that the enclosed ACL message payload is encrypted as defined in RFC822. In any case, the IETF has specified newer secure asynchronous messaging protocols such as Secure MIME (S/MIME) and Privacy Enhanced Email (PEM).

Note with any secure message protocol that there is a trade-off between the lack of confidentiality of information in the message envelope in order for message transport systems including network routers to be able to route messages, and the need for this information to be kept private in order to help prevent eavesdroppers knowing who is in communication with who. It also seems necessary to protect the integrity of the envelope information otherwise the delivery of message could be disturbed through corrupted envelopes. Finally, in some cases, different application requirements may require different combinations and strengths of message confidentiality and integrity, e.g., in some case clear-signed messages may suffice. The RFC822 model does not define any levels of granularity for the encryption and integrity where as the PEM and S/MIME models do.

Asynchronous message envelope encryption models set the encrypted field on a per message basis. There is no higher-level abstraction to specify message security for a group of messages such as on a per session or on a per interaction sequence or with respect to a policy. Synchronous message security models such as the Secure Socket Layer (SSL) message specification do contain the concept of a message stream. FIPA MAS can exchange messages using a FIPA specified MTP (Message Transport Protocol) such as HTTP over a lower level security protocol such as SSL. However, SSL is at a much lower level of abstraction than the level of the agent communication interaction sequences. For example, if a request-response interaction needs to be followed by a negotiation interaction, SSL has no knowledge of the message order inside an agent interaction and of the link between two or more agent interaction message sequences.

In addition, to considering how the confidentiality and integrity of individual messages and message sequences or interactions can be secured, we also need to consider where in the message generation and exchange, across multiple MASs (MMAS), the security is applied. For example the FIPA MTS specification considers that there are three options to define how an agent on a local MAS sends a message to another agent resident on a remote MAS (see numbered arrows in Figure 2):

1. Agent A sends the message to its local ACC Agent Communication Channel ACC (the main component of the Message Transport Service) using a proprietary or standard interface. The ACC then takes care of sending the message to the correct remote ACC using a suitable MTP.

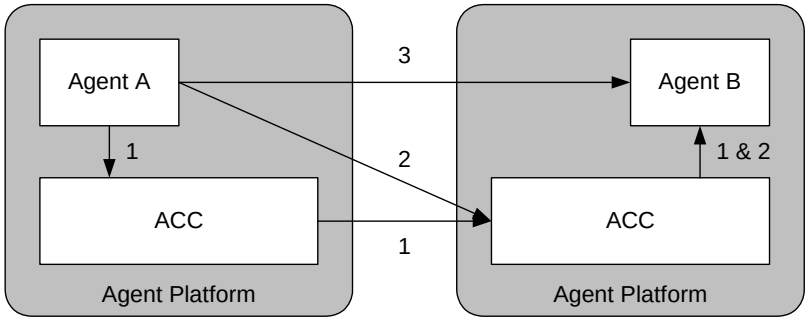

Fig. 2. Methods of Communication between agents on different Agent Platforms or MAS as defined in the FIPA Message Transport Specification. The numbers are explained in the main text. The ACC represents the Agent Communication Channel or Message transport Service

2. Agent A sends the message directly, using RMI for example, to the remote ACC, on the remote AP on which the receiving Agent, Agent B, resides. This remote ACC then delivers the message to B.
3. Agent A sends the message directly to Agent B, by using a direct communication mechanism. This communication mode is not covered by FIPA.

Security for the communication is not end-to-end in the sense of being application –to-application. Messages are encrypted in the message transport service in the Agent Communication Channel (ACC): the transfer of the messages to the transport layer service may be unencrypted. It is easy to eavesdrop on messages during their transfer from the agent to the ACC if they are unencrypted particularly if the message is transferred unencrypted to a remote ACC via interaction pattern 2 (Figure 2). Hence, interaction pattern 2 would not be secure.

3.4 Agent Name and Directory Service

In addition, to threats to the agent communication assets, threats to the name and directory service assets are also prominent in a MAS. As mentioned previously, the FIPA agent management specification [9] specifies name and directory services as agent-services, the MAS and DF respectively, but the FIPA abstract architecture defines the name and directory service more generally as NA-services or agent services. At this current time, the majority of FIPA MASs implement the name and directory services as agents. Note also that the Agent Management Specification combines the role of a name service with the role of a Agent life-cycle management, for loading, suspending etc. of agents, in an agent called the FIPA Agent Management Service (AMS). In practice, many FIPA AMS agents are used purely as a name service and the life-cycle management is performed by NA-services.

The main security issues are that the AMS and DF and other agents need to have access to authentication safeguards in order to verify an agent's identity and protect against masquerade threats. In addition the AMS and DF service agents need access control and authorisation safeguards to protect the name and directory information

that they store. There are no current FIPA specifications that define such safeguards. Hence, each MAS application is free to define these perhaps using other, non-FIPA, standards such as the use of X.509 certificates for authentication. The main issue is that within a MAMD environment, there will likely be interoperability problems between different MASs' use and interpretation of these authentication credentials. Many different kinds of authentication token are used in practice and they are application domain specific. The AMS registration specifies an ownership (a principal responsible for the agent) field in the service description frame of the agent management ontology – this ownership field has no integrity check and can be easily forged.

The agent management security specification was proposed [6] as a secure extension to the agent management specification. It defined a FIPA Agent Security Manager safeguard through which all communication passes; it enhanced the DF and AMS agent services and proposed fields in the transport envelope to set separate levels for confidentiality and integrity.

The strengths of the agent security specification model are:

The specification depicts abstractions for levels of privacy and integrity that are technology independent, i.e., they are specified as high, medium or low;

Message privacy is specified independently of message integrity. Multi-level model of confidentiality and privacy can be specified to support adaptive models of security, i.e., the agent can configure or reconfigure privacy and integrity according to application requirements or management policy.

This specification [6] however was never completed, implemented or used by the various MAS toolkits.

3.5 Review of FIPA MAS with Proprietary Security Systems

Security in agent mobility has been well researched, although no single or de facto standard has been developed. It is believed that mobile agents offer a greater opportunity for misuse and abuse [11]. This has led to the hypothesis that if we can solve the problems of mobile agent security then these solutions can be confidently applied to solve the security problems of other types (static) of agent system [12]. The main issues of security for mobile agents are that mobile agents must be protected from attacks by remote platforms and that remote platforms must be protected from attacks by mobile agents. As MASs of communicative agents reach out more into the untrusted heterogeneous environment of other MASs, communicative agents will likely face similar threats to those threats in mobile agent systems. There are however, important differences between a MAS of communicative agents and mobile agents: the protection of the agent code against code modification. Whilst this an obvious concern in mobile agent systems, it is not a major threat in MASs of communicative agents. Communicative agents are also more prone to communication threats than mobile agents. Multi-agent systems of communicative agents offer a comparable challenge to mobile agent systems, but to an extent, a different opportunity for misuse and abuse.

Whilst the current FIPA specifications contain minimal support for agent security, several researchers have reported adding security safeguards to FIPA based MASs.

They most often reported the addition of two key architectural safeguards: a secure channel to provide message privacy message integrity and a certification authority (CA) to provide authentication [13] [14] and [15].

Zhang et al. [13] have added security to the FIPA-OS MAS for mobile agents and communicative agents. The security service is implemented by two agents: a Secure Agent Communication Channel (SACC) agent to perform mutual authentication, and a Negotiator Agent to negotiate about the level of encryption to be used and to exchange symmetric keys for bulk encryption. Poggi et al. [14] report a security model for the JADE (Java Agent Development) FIPA MAS. Their approach uses a Certification authority, a distributed authorization model, and a secure channel based on SSL. Hu [15] has used the FIPA ACL combined with PKI for authentication and uses the SPKI (Simple PKI) model for authority delegation.

4. Some thoughts on future directions for FIPA MAS Security

The following are suggested as future research areas for FIPA:

Architectural Abstractions, services and design issues for MAS security;
Specifying multiple levels of security and the use of adaptable security;
Security, trust and Privacy;
Modelling security at the ACL level.

We have already presented a simple abstract model for MAS security, called the Asset Security Model, and we have discussed some of the limitations of current MAS implementations of such an asset model. This asset model needs further development, formalisations and reifications. Some of these further developments are touched upon in subsequent sections.

4.1 Adaptive Security profiles and policies

It is anticipated that dynamic adaptive model of security are needed in order to protect assets in a changing MAMD environment. Therefore one would have to define different groups of mechanisms that would be used in given situations. Some examples of adaptive security profiles could support:

The selection of non-confidential but integrity verifiable messages (i.e. readable by all but with certainty that they have not been tampered with), versus the use of encrypted as well as integrity verifiable messages (i.e. readable only by the intended recipient in addition to the certainty that they have not been modified).

The choice between public lookups of directory information (i.e. services and registered agents available for all to see), versus authenticated lookups (i.e. lookups restricted to some privileged agents).

The requirements for a minimal MAS security profile could include authentication; message privacy, detectable unauthorized message integrity violations and access control to key agent services.

The natural language type of security profile presented earlier, in Table 2 could be viewed as a private static policy that is specified during the requirements gathering

phase of the development of a new MAS application. It could also be expressed more formally and mathematically and perhaps used dynamically to reason and test whether safeguards are present. Profiles can also be made public. In a heterogeneous Multi-Agent, Multi-Domain (MAMD) world, published profiles would allow different domains to verify, negotiate and establish the necessary security to interoperate.

Security profiles are often associated with a specific policy or set of rules that could be expressed something like "if the system assets A and B are present and they are can be attacked by threat X then install and operate safeguard L to protect the asset against that threat."

In many distributed systems, the security policy is static and specified during the application requirements specification. Furthermore, the policy is often implicit in the sense that it is often specified in natural language form, by stakeholders such as users and developers, and then mapped to a particular system configuration for the operation of the system during the design and implementation phase. Were this policy to be represented explicitly, then it could also be used to dynamically manage the security of the system to adapt the safeguards in the face of changing threats and a changing operational environment. This is called policy-based security management.

Policies explicitly define the type of conditions a particular set of computational services will adhere to when operating in a particular context. This approach provides more openness to the service architecture as the computational services explicitly declare their intention to join a particular policy rather than this being implicitly defined within the communicative acts and protocols. Policies can be defined as a set of ontologies where the matching of policies can be done through a set of meta-constraint satisfaction rules. Examples of policies include policies for new-user registration, error handling, information sharing, delegation policy and control. The notion of policies can be applied to various concepts within an agent architecture, such as dynamic participation in teams [16]. More substantial work has been done in defining trust policies in [17].

4.2 Agent Communication Security

In previous sections, agent messaging has been viewed as a single asset that must be protected against integrity and confidentiality threats. But this is a very coarse grained representation. As application communication becomes more semantic and as standard interaction becomes richer to support the increasing number of open service spaces to support dynamic service mergers and demergers, we need a much more finely grained asset model of communication.

A more finely grained model of agent communication can be viewed as a set of four layers: transport level, speech-act or communicative level, ontology level and interaction protocol level, we examine the issues that should be considered with respect to providing security at each of these levels.

For the purposes of this discussion, a conversation is the set of related communicative acts (akin to a session) that comprise an interaction between two agents, and follows a given interaction protocol. A message contains a speech act and is associ-

ated with a single utterance within an interaction, and a message transport is the means by which a message gets from the sender to the receiver.

4.2.1 Transport Level issues

There is already much existing work in the area of message transport between processes, especially in the context of client-server models. Our security solution should take advantage of these as much as possible. For instance, it may be possible to fold transport-level security services under the umbrella of the transport service in the abstract architecture.

With that caveat, we also mention that sending messages between agents is not necessarily relegated entirely to some existing transport, so existing transport-level security may not necessarily cover agent message-passing. For instance, agents may use email or forward messages through gateway or proxy agents. Therefore, it is not clear that relying entirely on existing transport-level security is desirable. Finally, the lower down the network protocol stack, encryption occurs e.g., the IP layer, the less transparent and configurable it may appear to the agent. In addition, very low-level network layer encryption is not likely to be end-to-end.

4.2.2 Communicative Act issues

The addition of new communicative acts to access the security service has the advantage of simplicity. It has been proposed in several research papers, for example, He et al [17] have proposed adding new speech acts to KQML for apply-certificate, issue-certificate, renew-certificate, update-certificate and revoke-certificate. This approach could have been adopted for agent management in the agent management specification. The disadvantage is that FIPA has resisted adding service or application specific speech acts, for example for security, in order to keep the core set of speech acts generic and to a minimum. Rather than introduce new speech acts, an ontological approach is introduced as a powerful alternative approach.

Foner [18] was one of the first agent researchers to discuss the problem that many of the semantic models proposed for agent communication, require one agent to leak or reveal information about its internal state to another agent. For example, when one FIPA agent informs another agent that it is raining then the semantics of the inform communicative act require that the sender agent believe it is raining, and believe that the receiving agent does not yet believe it's raining and that after sending the message the receiving agent will come to believe it is raining. There is a trade-off in maintaining privacy versus using agent communication protocols that support rich knowledge exchange involving intentions, goals and plans. However, it is also possible to define some semantics for communication that do not depend on the sender and receiver sharing the same internal state.

4.2.3 Ontology Level

Making use of the existing FIPA speech acts and interaction protocols but referencing one or more security ontologies would minimize the changes to the existing ACL specifications to support security. It may be beneficial if FIPA seeks to reuse existing security schema from the mainstream computer network community. However, as

most security specifications are quite narrow, it is unlikley that a single security ontology could be specified - an approach is needed that can use multiple security ontologies. If the use of explicit ontologies are increasingly referenced at run-time in order to interpret richer messaging, threats to the integrity, availability and even the confidentiality of domain ontologies will become increasingly important.

4.2.4 Interaction Protocol Level

One key argument for providing security at the level at the interaction protocol level is that conversations naturally provide a scope for session keys. To wit, one natural paradigm is that an agent, wishing to interact with another agent in the context of some task, can authenticate itself to that agent; the agents can then share public keys that are valid for the duration of the interaction. This may be accompanied by the negotiation of policies at the interaction level – "This interaction takes place under the umbrella of this security policy … encryption method is …".

We note also that a given security implementation may have the potential to influence the interaction protocols themselves. For instance, if authentication becomes a part of every interaction among FIPA agents, this could either become some sort of a policy or could be embedded in the interaction protocols themselves. Also, the interaction with a security service may not naturally follow a pre-existing interaction protocol. Therefore new interaction protocols may need to be defined for such interactions (this may be true for services in general).

4.3 Security, Trust and Privacy

In the asset model of security, we define security in terms of profiles that specify safeguards to protect system assets against threats. The safeguards have been viewed as static localised objects, which we trust implicitly and totally. As safeguards become more complex, more adaptive and more non-deterministic, we may need to model the trust relation that we have with safeguards in a different way perhaps using social models. For example, conventional network authentication services tend to be trusted completely to supply, verify, and revoke authentication credentials but there have been a number of high profile cases published in which invalid credentials have been supplied and verified. We perhaps need to evaluate the relationship between the assets and safeguards using a probability and normalised model. We trust that safeguards and assets in the system have a normal behaviour and that they will seek to conform to this normalised behaviour.

Privacy issues and concerns can arise out of the use of personal data during single and multi-party interactions. Information privacy refers to the claims of individuals that data about themselves should generally not be available to other individuals and organisations, and that, where data is possessed by another party, the individual must be able to exercise a substantial degree of control over that data and its use. The main concerns relate to the disclosure, the lack of privacy of information, and misuse of the information by the holder of the individual's private information, and the corruption or misrepresentation of the information so that the person is misrepresented. Monitor-

ing of individuals without their consent, using information collected for one purpose for another and illegitimately passing on privacy personal information, are abuses of privacy information disclosure and corruption are further concerns. Hence, privacy information needs to be safeguarded. Immediately, we can deduce that simple access control coupled to data encryption are not panaceas to meet these challenges. To guard against the latter threats, privacy enhancing safeguards and privacy information usage policies need to be used to minimise these threats.

5. Conclusions

If multi-domain services interactions involving semantic communication, richer interaction patterns such as negotiation, personalized access and local context awareness become more routine, the need for more sophisticated security models for agent based communication becomes necessary. These are needed to support the legal concerns for data protection, the use of personal preferences, social and moral concerns, and to support the general security requirements for e-business.

6. Acknowledgements

We thank all our colleagues within FIPA who have contributed to or reviewed the security white paper. We also thank people for their response to the FIPA Security WG Request for Information, issued to the agent and security community. In addition, the FIPA Security WG wishes to thank the membership for its input during the FIPA meetings and to others for contributions to the email list. The views expressed in this article are those of the authors. Stefan Poslad was partly supported by the EU (IST-1999-20147) CRUMPET project.

7. References

1. Schneier B. Secrets and Lies: Digital Security in a Networked World, Wiley, (2000).
2. Nass C and Reeves B. The Media Equation: How People Treat Computers, Televisions, and New Media as Real People and Places. Cambridge University Press, (1996).
3. Falcone R., Singh M., and Tan Y. (Eds.) Trust in Cyber-societies: Integrating the Human and Artificial. Perspectives, LNAI 2246 Springer, (2001).
4. FIPA MAS Security White paper, reference f-out-00113, http://www.fipa.org/repository.
5. FIPA, The Foundation for Intelligent Physical Agents, Home Web-page. http://www.fipa.org.
6. FIPA 98 Part 10 Version 1.0: Agent Security Management Specification (obsolete). http://www.fipa.org/repository/obsoletespecs.html
7. FIPA Abstract Architecture Specification., Version J, http://www.fipa.org/repository.
8. FIPA Agent Message Transport Service Specification. http://www.fipa.org/repository.
9. FIPA Agent Management Specification. http://www.fipa.org/repository.

10.Crocker D.H. Standard for the format of ARPA Internet Text Messages. IETF Request for Comments 822.
11.Jansen W and Karygiannis T. Mobile Agent Security, National Institute of Standards and Technology Special Publication 800-19 (August 1999)
12.Ghanea-Hancock R, Gifford I. Top secret multi-agent systems. 1st Int. Workshop on security of mobile multi-agent systems (SEMAS-2001), 5th Int. Conf. Autonomous Agents, Montreal, Canada (2001).
13.Zhang M, Karmouch A and Impey R. Towards a Secure Agent Platform based on FIPA. Proc. MATA 2001. Springer-Verlag. LNCS, (2001), Vol. 2164, 277-289.
14.Poggi A, Rimassa G and Tomaiuolo M. Multi-User and Security Support for Multi-Agent Systems. Proc. of WOA 2001 Workshop, Modena, (Sep 2001).
15.Hu Y-J. Some thoughts on Agent Trust And delegation. Proc. 5th Int. Conf. on Autonomous Agents, AA2000, Montreal, (2000) 489-496.
16.Charlton P. and Cattoni R. Evaluating the Deployment of FIPA Standards when Developing Application Services", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, No. 3, (2001) 551-577.
17.He Q, Sychara K, Finin T. Personal Security Agent. KQML-based PKI. Proceedings of (AA'98) Autonomous Agents (1998).
18.Foner LN. A security architecture for multi-agent match-making. Proc. ICMAS (1996).