FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA MAS Security white paper

Document title	FIPA MAS Security white paper		
Document number	f-out-000113	Document source	FIPA Security WG
Document status	Final	Date of this status	2002-04
Contact	security@fipa.org		
Change history			
2002-04-30	Version 1.6: final version		

Geneva, Switzerland

Notice

Use of the technologies described in FIPA's specifications may infringe patents, copyrights or other intellectual property rights of FIPA Members and non-members. Nothing in this document should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of FIPA's specification to determine first whether part(s) sought to be implemented are covered by the intellectual property of others, and, if so, to obtain appropriate licenses or other permission from the holder(s) of such intellectual property prior to implementation. This document is subject to change without notice. Neither FIPA nor any of its Members accept any responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of FIPA's specifications.

Foreword

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. This occurs through open collaboration among its member organizations, which are companies and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested parties and intends to contribute its results to the appropriate formal standards bodies.

The members of FIPA are individually and collectively committed to open competition in the development of agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual firm, partnership, governmental body or international

organization without restriction. In particular, members are not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by virtue of their participation in FIPA.

The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process of specification may be found in the FIPA Procedures for Technical Work. A complete overview of the FIPA specifications and their current status may be found in the FIPA List of Specifications. A list of terms and abbreviations used in the FIPA specifications may be found in the FIPA Glossary.

FIPA is a non-profit association registered in Geneva, Switzerland. As of January 2000, the 56 members of FIPA represented 17 countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found at http://www.fipa.org/.

Summary

Distributed multi-agent systems propose new infrastructure solutions to support electronic service access and interoperability. Although agent technology is attributed with many properties, even a key focus on just two aspects, autonomy and communication, can support reconfiguration, scalability, loosely-coupled service definitions and flexible service delivery and interaction. These can in turn support peer-to-peer services, negotiation and personalised services.

Frequently, although security is a central issue, it is treated as an orthogonal infrastructure that can be plugged in after development is complete. It is possible under certain conditions and applications to treat security in such a dynamic way but this leads to limited point solutions and does not support the increasing demand for openness that is seen in the use of the Internet, in general.

If services such as automated negotiation, personalized access and local context awareness are to be supported by agent technology then security becomes necessary. It is needed to support: the legal concerns of data privacy, the use of personal preferences, social and moral concerns, trust issues and the general security requirements for e-business.

This white paper discusses the issues for integrating agents and security. It reviews some current concerns, illustrates the main issues using simple case studies and outlines the relationship between security, trust and privacy. There is a coarse grained analysis of the FIPA agent services typically reified as an agent platform in order to understand the relations between threats, safeguards, and mechanisms. Finally this article outlines the direction of future work in order to specify security for MASs. The appendix contains a glossary of some security terms used in this article.

Table of contents

1 Scenarios and Requirements for MAS Security

1.1 MAS Security for Open Service Spaces

The e-business world is being driven to create and enhance technologies to support dynamic services, automated interaction, rich information exchange and tailored solutions. These kinds of environments are generating complex, semantic, collaborative and competitive behaviours that are being substantially researched and studied within the agent community. Hence Multi-Agent Systems (MASs) are an important technology that is being closely evaluated and deployed in order to cope with the growing needs of automated coordination in heterogeneous, dynamic environments in the e-business world.

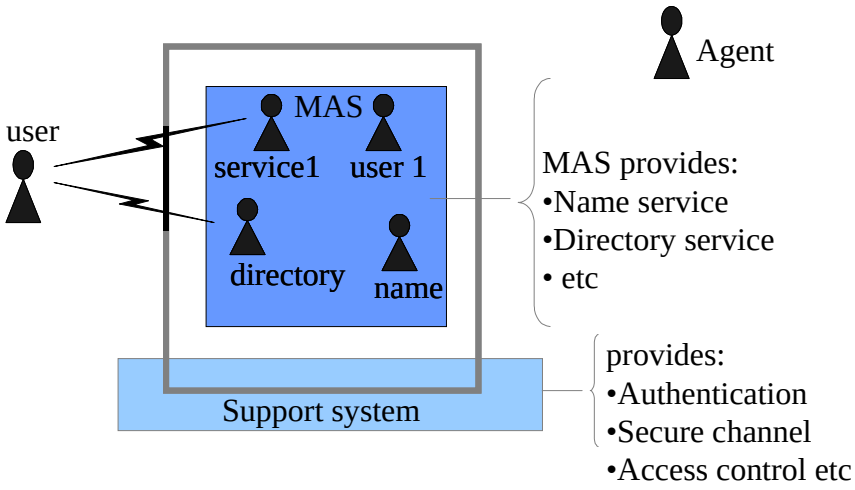

Figure 1: classic protection of a distributed system

However, only very specific areas within the Internet space offer advanced security solutions to protect against malicious attacks: these are typically centralised closed systems (see Figure 1) that rely heavily on human supervision and control. Users are becoming increasingly aware of the problems of using Internet systems, e.g., experiencing fraudulent transactions even if they have never used a particular on-line financial service. The security of communication within an increasingly ubiquitous Internet is still a very open and critical issue.

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, which has not received extensive attention until very recently.

Addressing such a complex problem requires understanding the requirements and properties of systems that operate in these environments and how these affect security. It is also necessary to define how agents and their peculiar properties such as autonomy, social ability and pro-activeness can work with the underlying infrastructure to provide security. Finally, it is important to foresee and try to address a new generation of security issues that will arise as automated, dynamic behaviour and peer-to-peer interaction become more pervasive and as agents become promoted to act on behalf of humans.

Typically, agents in multiple domains, and hence in multiple MASs (MMASs), will need to interact (Figure 2) over public networks, e.g., a supply chain and travel service brokerage. Securing services in distributed domains is complex. Although it is possible to secure each part of the distributed system this hinders multiple-domain MMAS interaction.

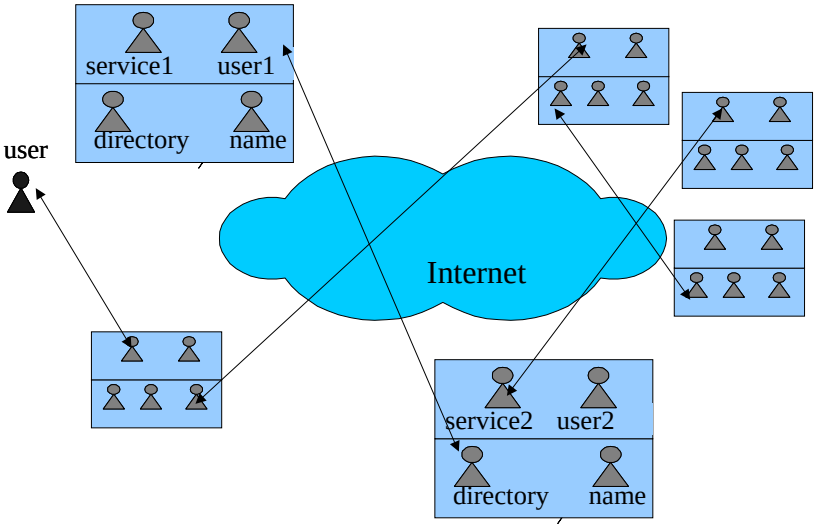

Figure 2: the problem of securing activities in multiple domains

In some service spaces for agent enabled e-businesses, the network infrastructure may be insecure, different service components may be sourced from multiple vendors and the service and network infrastructure may also be segmented or layered and sourced from multiple providers. We call this an open services space (OSS). Examples of an OSS include the Internet, a Virtual Private Network, a Value Added Reseller (VAR) service architecture, a supply-chain and an open marketplace.

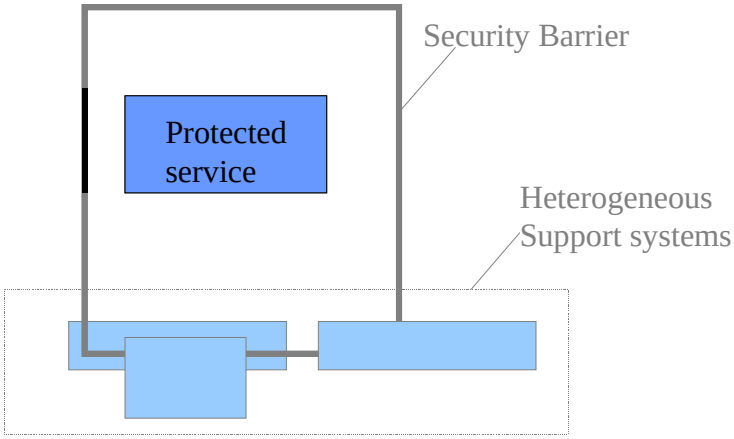

Figure 3: multiple support system components may inadvertently combine and weaken the security (depicted by breaks in the security barrier).

In an OSS, we could secure each component but then combining different components may compromise the security of the individual parts (figure 3). For example, if we introduce a service component that requires authentication and then introduce another component that buffers the credentials or identities insecurely, then we introduce new insecurities. If we could identify the conditions under which, service components offers protection, i.e. its security policy, then we may be able to avoid combinations of service components that introduce “emergent insecurities” or at the very least be able to evaluate the advantages and disadvantages of different architectures.

There is a further complication, security policies, for example, the rules that govern access control to resources and the conditions under which message encryption is used, are bounded: they are designed to apply only within a limited space called a domain. However, frequently policies that apply in one domain may not apply when that domain becomes linked to another domain. For example, conflicts can occur when a company has a policy to withhold information becomes part of a domain whose policy is to make that same information public.

To summarise, a vital problem space for the use of secure MASs are e-business OSSs that are characterised by:

- Heterogeneous service components from multiple providers;
- Dynamic service aggregation;

- Information distributed across possibly insecure environments;
- Multiple autonomous domains that may become interlinked and loose some of their autonomy.

1.2 Security, Trust 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 within it [NAS96]. 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 creating a high-level concept of trustworthiness requires support for generic concepts of trust, security, and privacy, within a multi-agent systems architecture. Trust, security and privacy can be defined as follows:

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

Security: is a set of physical realisations that reduce the risk of danger or 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. These are introduced in order to manage the authentication and authorisation credentials – these services are fully trusted by the users of the service. 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 and services that a person or community places a high-value on and that these items are respected as such. Often privacy utilises both security and trust.

1.3 Issues in totally controlling security outside the MAS

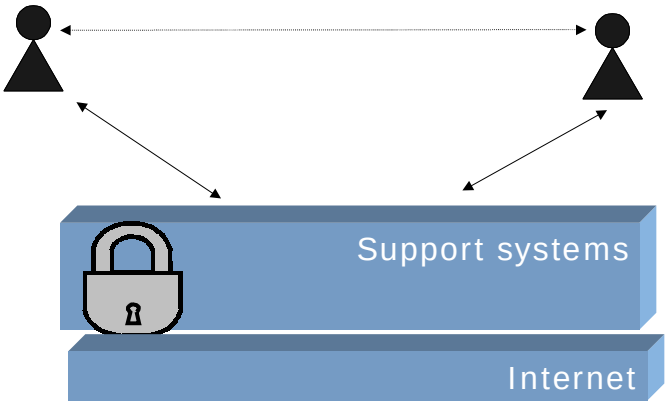

Figure 4: The MAS delegates security to the support system

The classic way of securing a (MAS) service or application is for the support system or infrastructure to provide total security for the agent system, that is, security is outside the scope of the MAS "layer". This can also be thought of as the MAS layer delegating the security action totally to an external component (Figure 4). Let's assume an agent uses the support system to provide privacy, integrity and authentication for interaction with another agent. When attacks succeed, or failures such as software crashes occur in the support system – this security is violated but the agent may remain ignorant of this. This may be because:

- Support systems and infrastructures may intentionally mask this reduction in system security as it perhaps wants to protect its reputation;

- The support system is not a stakeholder in the interaction - it thinks that such lapses should not concern the agents;
- Support systems are frequently layered, one system may provide the core support operation but other systems act as Valued-added Resellers (VARs) of these services. Frequently, services users can never ascertain who is responsible and who is liable as the core operational services and the VARs apportion responsibility to each other.

Thus, in some situations, "blind trust" or out-sourcing of security may introduce new security problems. This suggests that security offered by support systems to agents needs to be monitored.

1.4 Degrees of agent control of security

There are different degrees to which agents can manage security (Figure 5). Agent management of security may range from no control in delegating security management to the non-agent software infrastructure, to finely grained control in delegating security.

An overriding concern is that the management of security is complex. The properties of agents such as social ability, autonomy, proactivity and inbuilt reasoning provide a good basis for managing the complexity of system security.

Making open distributed information systems secure is a challenging endeavour. Whilst cryptography provides the means for establishing secure peer-to-peer communication, it requires complex management of keys. Furthermore, although pure cryptography may offer the best protection, for example through encrypting everything including the transport envelope containing the sender's address, it is in contradiction with the even more crucial requirement of authentication, i.e. making sure that communication is transparent and indeed from the sender and to the intended recipient.

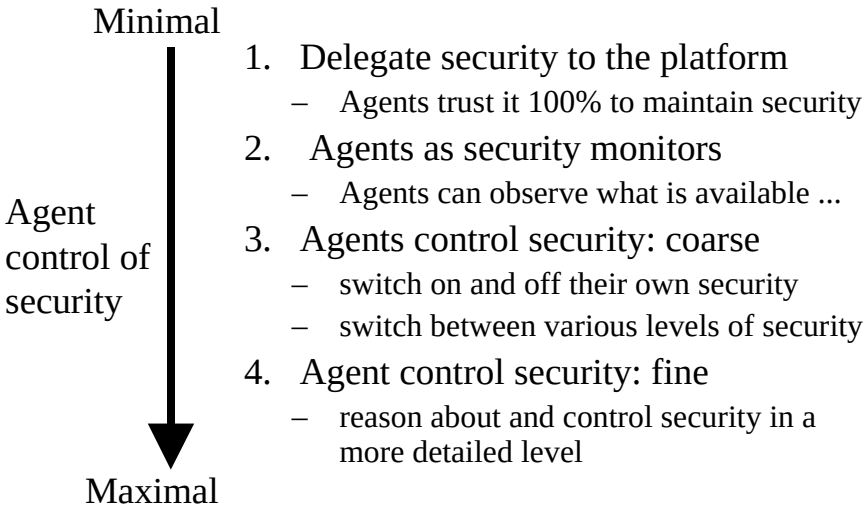

Figure 5. Agent control of security from ranging to minimal control to maximum control

By moving communication control to a higher level of abstraction, agents offer ways out of this dilemma. Cryptographic communication can be embedded at a lower level in the protocol stack for efficiency while being controllable at a higher level. Authentication can make use of agents' persistence over time and refer to past interactions within a wider body of experience, similar to the way people do. Agents can interact with agents to assess the behaviour of a particular agent, or even the infrastructure itself, to build up a distributed model of security. Finally, depending on the degree of autonomy that is available, agents could run their own security checks and independently configure their own security depending on their application and environment contexts.

1.5 Scenarios and Use-cases

In this section, we present a series of simple security-related scenarios to illustrate some of the pertinent issues and to produce requirements for MAS security (see Table 1).

Scenario	Security Requirements
Publisher / directory	Authentication, authorisation, 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
Personalisation and privacy service scenario	Privacy & integrity of user preferences, and service capabilities, authentication of owner, authorisation, policy integrity, privacy & trust
Mobile agent application scenario	Agent and host integrity and access control

Table 1: Overview of security issues in different scenarios.

For each of these scenarios, a selection of threats is described. This is not meant to be an exhaustive threat analysis in each case - the aim is to cover a wide spectrum of threats across all scenarios.

1.5.1 Publisher / Directory Scenario

Let's assume that we have five agents:

- Agent A wants to buy fruit and ice cream;
- Agent B sells fruit;
- Agent C sells ice cream;
- Agent D is a malicious agent.
- FIPA DF agent [FIPA00023] acts as a service directory for services by different vendors. All directory entries in the DF are public. Agents B and C register their services, i.e., "selling fruit" and "selling ice cream", with the DF.

Given the current FIPA specifications, agent D can:

- Pretend to be agent C and change C's service description at the DF, declaring that C now sells wine instead of ice cream. In this way agent C will not be able to make any profit since it will be asked to provide a service that it is not able to support and nobody will ask it for the real service it is able to provide;
- Pretend to be agent A and place an order with agent B for a 100 kG of plums. Therefore, agent B sends 100 kG of plums and the bill to agent A. Who is going to pay the bill? What about the plums?
- Pretend to be agent B and gets paid for providing a service that it does not intend to honour;
- Pretend to be a multitude of customers and overload B and C with a deluge of fake offers to be processed resulting in valid service offers becoming delayed and perhaps causing them to be subsequently withdrawn.

In this scenario, the main security threats are:

- Lack of authentication: an agent can masquerade as another agent;
- Lack of authorisation: one agent can change another agent's published service description;
- Denial of Service (DoS) Attack: a masquerading agent can change another agent's published service description thereby disrupting the normal service provision of that agent.

1.5.2 Courier / Broker Scenario

Let assume that we have three agents (Brokerage is discussed in more detailed in [FIPA00033]):

- Agent A is a buyer;

- Agent B is a seller;
- A courier agent (broker) agent acts as an intermediary for all messages between the seller and the buyer.

If all discourse between A and B is via the courier agent, the following security related problems can occur:

- The courier agent can open up messages between A and B and observe their contents;
- The courier agent can modify the messages between A and B;
- The courier agent can insert messages from other parties to masquerade as A or B. For example if the courier was an estate agent (a broker that sells houses), it could hypothetically say that "A has pulled out of a deal to buy the house", thus putting pressure on B to possibly sell at a lower-price;
- Agent A can deny having sent a message, the courier can deny having been called to deliver it, agent B can deny having received a sent message.

In this scenario, the main security concerns are:

- Lack of privacy: content of messages can be monitored by unauthorised entities;
- Lack of integrity: messages can be misrepresented or corrupted;
- Lack of authentication: an agent can masquerade as another agent;
- Repudiation: an agent can deny that messages have been received or sent.

1.5.3 Task allocation Scenario

In the contract net protocol (see [FIPA00029]) a manager agent wishes to have some task performed by one or more other contractor agents. The manager specifies the task and any conditions it places upon the execution of the task, issues a call and solicits *proposals* from other agents. Agents receiving the call for proposals are viewed as potential *contractors*, and are able to generate proposals to perform the task. They may also be able to sub-contract the task to another contractors, behaving as a manager in this level of interaction. Alternatively, the contractor may *refuse* to propose. Once the manager receives back replies from all of the contractors, it evaluates the proposals and makes its choice of which agents will perform the task. One, several, or no agents may be chosen. The agents of the selected proposal(s) will be sent an acceptance message, the others will receive a notice of rejection. The proposals are assumed to be binding on the contractor, so that once the manager accepts the proposal the contractor acquires a commitment to perform the task. Once the contractor has completed the task, it sends a completion message to the manager.

For the interaction between the manager and the contractor agents, the following security related problems could occur:

- Contractors can deny having received a rejection or acceptance;
- A contractor can modify details of the contract when it is passed on to a sub-contractor.

In this scenario, the main security concerns are:

- Lack of integrity: a manager or contractor can misrepresent or corrupt messages
- Non-repudiation: the manager or contractor can deny that an event such as a message transmission or message delivery has occurred or introduce a spurious event such as an unsubstantiated message;
- Lack of authority delegation and lack of privacy: an agent such as a manager cannot control how second parties protect confidential information on to third parties.

A secondary concern is that there is a:

- Lack of a framework to establish and fix an agreement and set the conditions under which the agreement can be modified.

1.5.4 Multiple service domains scenario

Agents operating within one domain may require access agents in different domains. We wish to consider how a service trusted to operate in one domain becomes trusted to operate in new domains and to consider what security threats and trust abuses can occur.

For example, consider a travel service [FIPA00080], a travel broker can access and use many different types of travel services such as flight reservation services, hotel room reservation systems, train travel reservation services, travel insurance service and so on. Some of these services are location specific such as train travel whilst others may be global such as airline reservation.

Rather than model the travel assistance service within a single domain, we can model a travel service as a distributed set of geographical and application specific domains that may interoperate, e.g., travel service domains such as a global flight-reservation and a local train reservation service. We must consider how services in these different domains can be set-up to interact.

Let's assume a travel broker is already registered to operate flight reservation services. We can say it is trusted to operate a flight reservation service. Let's consider how it interoperates and how it becomes trusted to operate train reservation services.

There are at least two basic "bootstrapping" patterns to describe how an agent trusted in one domain becomes trusted in another domain: there may be an inter-domain agreement or each agent must register and elevate itself to become trusted in each domain it operates in. The former option requires information about the trustworthiness of agents in one domain to be transferred to another domain. For example, because the travel broker is licensed or registered to operate in the airline reservation domain, there may be an inter-domain agreement that allows it to operate as a fully-fledged train reservation service provider and without having to authenticate itself to its peers and to the establishment in the new domain. There are several problems that can occur:

- Providers may try to mis-represent themselves in different domains, e.g., mask a loss of status in one domain such as a downgrade in financial status in another domain;
- The transfer of an agent's identity and trustworthiness information between domains may get falsified;
- The issue of what it means to transfer trust established in one domain across to another domain is not defined.

1.5.5 Personalisation and privacy service scenario

In this scenario, there are the following agents:

- A personal agent A that holds a person's preferences and characteristics such as tolerance to drugs, gender etc.;
- A doctor service agent B 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 that medical condition in the patient;
- Other agent services, C, D, e.g., hospital services, may be used by agent B to carry-out its service;
- Other personal agents E and F may also talk with agent A to find out about information about C's service.

The following security problems can occur:

- 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 switched 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 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 B to treat A but B gets replaced by another instance of the doctor agent.

Decker et al [DEC97] define nine basic interaction patterns for revealing and sharing service capabilities and user preferences amongst users, providers and mediators. This type of scenario also introduces a need to define, distribute and uphold policies for protecting the privacy of information such as user preferences and provider capabilities. There is a requirement to protect the identity of the human principal that is associated with an agent. In addition, this scenario illustrates the need to deal with multiple certification authorities. In this scenario, the personal agent or the hospital agent may need to certify that the doctor instance agent has the qualifications to offer a doctor service. A hospital needs to be certified to offer facilities for doctors to use. Finally, there are legal issues for privacy that need to be addressed, see for example the EU PISA project [PISA].

1.5.6 Mobile Agent Application Scenario

Drake & Morse [DRA01] describe some of the pertinent issues of mobile agent security in a scenario called the "Byzantine Princes". This concerns a dying king who has four sons who are princes. He wishes to distribute his inheritance among them. Each of the princes has his own principality. Each principality is run ruthlessly, and the princes don't dare leave their respective castles, for fear of being killed by their own subjects. The King has planned a test to decide who will get the inheritance. The princes will be grouped by into teams of two, and the teams are to play a game of chess. Each side is to alternate who makes their move. The team that wins will split the old king's fortune evenly; the losers will get nothing. In this scenario, we can regard the:

- The Dying king as the owner and home for the mobile agent;
- The chess-board, the caravan and the associated personal as the mobile agent;
- The castles of the four princes as the remote hosts, which the mobile agent visits.

Drake & Morse [DRA01] identify the following security concerns and threats:

- Denial of service
    - A caravan may be destroyed while in transit;
    - A prince may have the caravan destroyed while it is at his castle.
- Lack of Integrity
    - The board position carried by a caravan could be changed while in transit;
    - The caravan personnel may be exchanged with people with a different mission, including ones who wish to misuse the access rights of the caravan, or want to modify the legitimate outcome of the game;
    - A prince may attempt to change the board position before his move;
    - The caravan personnel may harm the prince.
- Masquerade
    - Imitators of the caravan may arrive at a castle;
    - A prince could send a imitation caravan ahead to see what the next round of moves would be, and then give his illegal chess move to the real caravan, and send it on. This can also include sending this to the king to mislead him that a certain prince has won the game;
    - Imitation castles may be set up to deceive the caravan into visiting the wrong prince, or someone who isn't a prince at all;

- A prince could send out a duplicate caravan as well as the real one, so that multiple games are being played. At a later move, the prince or his brother may destroy the caravan with the less favourable chess position.

- Lack of Privacy:
  - A prince's subjects may lose confidence in the prince if his ineptness were revealed by disclosure of the board's status.

- Trust abuse:
  - The caravan personnel could be bribed to act improperly;
  - A prince may attempt to make an illegal move;
  - A prince may attempt to never make his move.

- Non-repudiation:
  - A prince may falsely claim that an illegal move has been made to try to force a number of moves to be taken back.

The principle threats in the mobile agent scenario are that the (mobile) agent's infrastructure (the remote hosts) must be considered hostile (to the home host) and that the different agent hosts may be hostile to each other. As a result the integrity of the agent itself, its state and behaviour, may become compromised. Another view of mobile agents is that they represent a complex message ("the intelligent messenger") that is sent around to a group of static agents (the remote hosts or receivers). The message integrity may be compromised by any of the receivers in order to mislead other receivers or the sender.

2 FIPA Agent Security: (1997-2002)

In this section, we review the security models that are described within selected FIPA specifications. We also consider how the FIPA agent specifications have been used by others to develop agent security applications.

At this time (2002), FIPA does not have a strong agent security model mainly because FIPA felt that the issue of where and how to add security to FIPA ACL-based systems, needed much more debate. But just because security isn't specified at the ACL level, it doesn't mean that agents can't have security – security can be accessed at a non-ACL level.

The current abstract architecture specification [FIPA0001] 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 listed several key requirements for security, these are:

- **Identity**. The ability to determine the identity of the various entities in the system. By identifying an entity, another entity interacting with it can determine what policies are relevant to interactions with that entity. Identity is based on credentials which are verified by a Credential Authority.

- **Access Permissions**. Based on the identity of an entity, determine what policies apply to the entity. These policies might govern resource consumption, types of file access allowed, types of queries that can be performed, or other controlling policies.

- **Content 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. Digitally signing data and then having the recipient verify the contents are unchanged often accomplish this. Other mechanisms such as hash algorithms can also be applied.

- **Content Privacy**. The ability to ensure that only designated identities can examine software, a message or other data. To all others the information is obscured. This is often accomplished by encrypting the data, but can also be accomplished by transporting the data over channels that are encrypted….".

Security issues are of concern in the following services such as:

- Message transport service: [FIPA0067];

- Agent management service: [FIPA00023];
- Security support services: [OC00019].

Each of these will be discussed in turn.

2.1 Message Transport Service

The current FIPA ACL semantics guarantees that the exchange of messages between agents is coherent with what agents believe, desire and intend to do, but this is effectively true only under the main assumption that agents are truthful. However, FIPA cannot (and should not) prevent agents to `be economical with the truth'. Therefore, given that the semantics of FIPA ACL by itself does not give guarantees about agents' honesty, standard specifications should provide a way to reduce the effect of malicious agents (or malicious platforms) by supplying transport level mechanisms to encrypt messages and to verify their integrity.

The FIPA Message Transport Service [FIPA0067] specifies that an agent has three options when sending a message to another agent resident on a remote agent platform (see numbered arrows in Figure 6):

1. Agent A sends the message to its local ACC using a proprietary or standard interface. The ACC then takes care of sending the message to the correct remote ACC using a suitable Message Transport Protocol or MTP;

2. Agent A sends the message directly to the ACC on the remote AP on which 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.

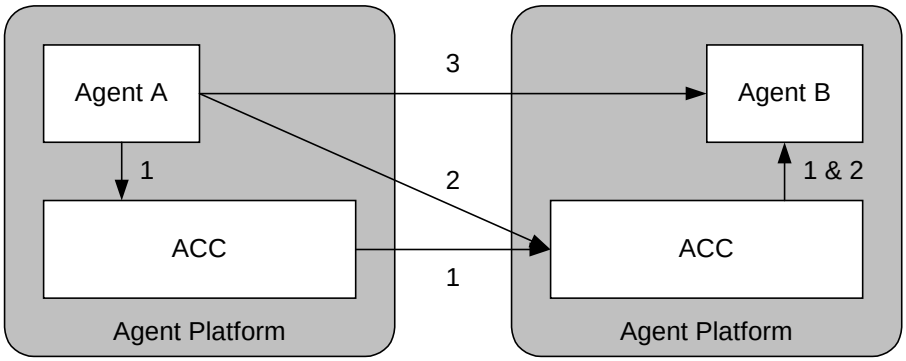

Figure 6. Methods of Communication between Agents on different Agent Platforms via the the Agent Communication Channel (ACC) as defined in the FIPA Message Transport Specification. The numbers are explained in the main text.

The MTS transport specification [FIPA0067] adds a header to each ACL message for transport (Table 2). This header specifies the use of an encryption field that references RFC 822 [RFC 822] to encrypt ACL messages. RFC 822 does not support MIME, header integrity, header privacy and is not supported by FIPA agent platforms. RFC 822 has been superseded by newer IETF specifications such as the secure MIME specification [RFC2633], which unlike RFC 822, supports MIME together with authentication, message integrity, non-repudiation of origin (using digital signatures), privacy and data security (using encryption).

Frame Ontology	envelope FIPA-Agent-Management			
Parameter	Description	Pres-ence	Type	Reserved Values
to	This contains the names of the primary recipients of the message.	M	Sequence of agent-identifier	
from	This is the name of the agent who actually sent the message.	M	agent-identifier	
…				
payload-length	This contains the length of the message body.	O	String	
payload-encoding	This contains the language encoding of the message body	O	String	US-ASCII, ISO-8859-1,ISO-8859-9, UTF-8, Shift_JIS EUC-JP, ISO-2022-JP ISO-2022-JP-2
…				
encrypted	This contains information indicating how the message body has been encrypted.	O	Sequence of `String`	See [RFC822]
intended-receiver	This is the name of the agent to whom this instance of a message is to be delivered.	O	Sequence of agent-identifier	
…				
transport-behaviour	This contains the transport requirements of the message.	O	(Undefined)	

Table 2. Part of the Envelope specification in the FIPA Message Transport (MTS) Specification [FIPA0067]

The message envelope encryption model sets 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. A common technology used to support secure sessions, at this time, is SSL [RFC2246].

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 6). Hence, interaction pattern 2 would not be secure.

2.2 Agent Management Service

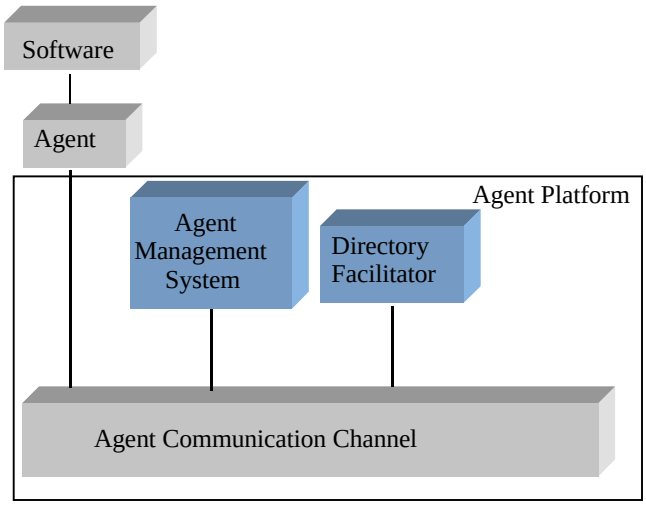

Figure 7. The Agent platform as defined in the Agent Management Specification

The current FIPA agent management specification [FIPA0023] defines the concept of an agent platform (Figure 7) - a physical infrastructure in which agents are deployed. As agents in MASs tend to use facilitator agents to discover and maintain the knowledge of other agents that they need to interact with, [FIPA0023] specifies the use of a DF (directory service agent) and an AMS (name service and agent life-cycle management service agent). The current FIPA00023 does not specify any security at the ACL level. The AMS and DF do not specify any credentials to verify an agent's identity; the AMS and DF directories have no access control at the ACL level.

In addition the AMS registration specifies an ownership (a principal responsible for the agent) field in the service description frame of the agent management ontology [FIPA0023] – but the binding of the owner to agent has no integrity check at the ACL level.

.

2.3 Agent Security Support Service

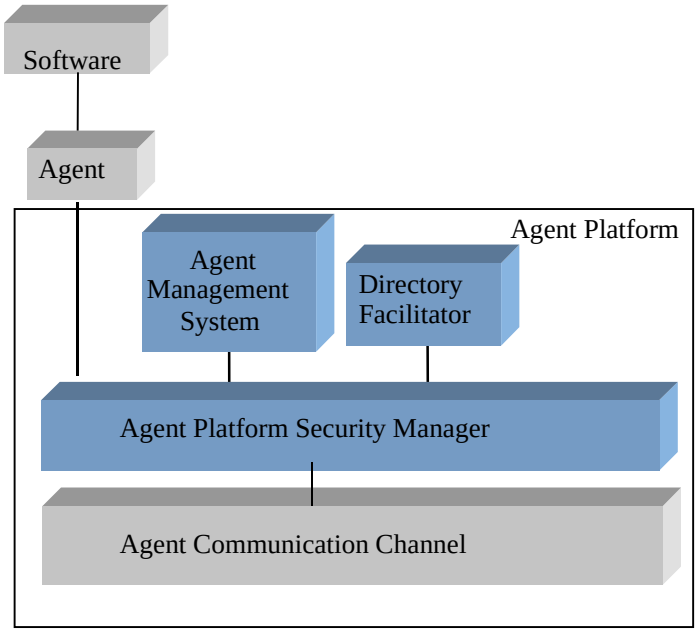

Figure 8. The agent platform as defined in the agent security management specification

In 1998, FIPA first became active in specifying agent security at the ACL level in the agent security management specification [OC00020]. Because this work was incomple and partly because FIPA felt that the issue of where and how to add security to FIPA ACL-based systems needed much more debate, the agent security management specification was obsoleted and the security "hooks" in the agent management specification were removed. However, it is still worthwhile examining the obsolete specification in order to review its security models (see below).

The specification defines:

- confidentiality mechanisms for keeping message private over a public network;
- integrity mechanisms for ensuring data has not been tampered with during transfer;
- authentication mechanisms to ascertain the identities of agents.

These mechanism are specified on a per message basis by the agent service user by setting fields in the envelope of the ACL language construct (see below).

2.3.1 The OC0020 security enhancements to the AMS and DF

This model also enhanced the roles of the AMS and DF agents and introduced an entity called the APSM (Agent Platform Security Manager) to be specifically responsible for maintaining security policies.

The enhanced AMS is primarily responsible for authenticating agents within the agency and uses private and public keys for authentication. Key pairs need to be exchanged between the AMS and agents and the agent owner of the private key and are stored in the AMS and in the particular agent.

The agent services define the security they support by specifying additional parameters in the service descriptions they register with the DF such as certificates for authenticating public keys for the agent service and the human owner and the confidentiality encryption technique.

Although, key management is described for authentication, key management is not defined for some of the confidentiality mechanisms defined in the security model.

2.3.2 The OC00020 Message Transport Envelope

At the communication level some preliminary suggestions for secure ACL communication have been made in [OC00020]. Particular attention is devoted to the 'envelope construct', since the transport level

protection relies on the information specified within the envelope. The main idea is to specific security fields for confidentiality and integrity so that an agent can request security services, but the responsibility of encapsulating the messages lies with the message transport mechanism. An example of a `secured' ACL message using this model is given in Figure 9.

At the agent communication level this implies the design of a common standard ontology that should be able to capture and define all the main terms and definitions related to the confidentiality, the integrity, the authentication and the non-repudiation mechanisms provided by the agent platform.

```
(letter
:envelope (
            :destination(…)
            :return-address (…)
            :confidentiality high
                 :integrity high )
:message
          (refuse
                  :sender …
                  :receiver ….
                  :ontology ….
                  :content …..)
```

Figure 9: an example of a `secure' ACL message envelope

The strengths of the agent security defined in this message envelope model include:

- 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 underpin adaptive models of security, i.e., the agent can configure or reconfigure privacy and integrity according to application requirements or management policy.

2.4 Review of published reports of MAS security

Security has been most avidly researched for the mobile type of agent system, based on the papers retrieved by search engines using the terms agent and security. This is perhaps because mobile agents simply offer a greater opportunity for misuse and abuse [JAN99]. There is also 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 of agent systems [GHA01].

The research on mobile agent system security has counterparts in non-mobile agents systems such as conventional client-server system security. For example, direct attacks on the code integrity of a mobile agent by an untrusted foreign host environment can be equated to integrity and DoS attacks by an untrusted remote agents, e.g., they can construct messages to cause the receiving agent's message handler to fail. 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 MASs of communicative agents and mobile agents: the protection of the agent code against code modification whilst being an obvious concern in mobile agent systems 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.

Even although, the current (non-obsolete) FIPA specifications contain minimal support for agent security, several researchers have reported adding security to FIPA based MASs. MASs most often reported the use of encryption-based mechanisms to protect systems. Two key architectural elements are added: a secure channel to provide message privacy and a certification authority (CA) to provide authentication [HE98a], [POG01], [Zha01] and [HU01].

He et al [HE98a], [HE98b] propose adding agent security to the Retsina MAS architecture - at the time of the published reports, the addition of security was work in progress. In Retsina, the authentication service is performed by a type of middle agent called the Security Agent. Agents in Retsina such as the Security Agent consist of modules. The security agent consists of three modules: AgentEditor, Planner and security module. The AgentEditor is used to define the properties of and policies for operating an agent – for the case of the security agent, policies such as mutual vs. one-way authentication and the use of digital signatures can be specified. The Planner constructs and implements these protocols, for example, for certificate update and certificate revocation. The security agent can manage multiple types of user-definable public key certificate chains including standard X.509 chains.

Zhang et al [Zha01] have added security to the FIPA-OS MAS for mobile agents and communicative agents. The security service is modelled 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. There is also a central certificate authority (CA) that is trusted by each MAS and issues authentication certificates to MAS. In this system, three security processes take place: mutual authentication between multiple MAS, assigning authorisation credentials for agents to access resources within a MAS, negotiated configuration of a secure channel between different MAS.

Poggi et al [POG01] report a security model for the JADE (Java Agent Development) FIPA MAS. Their approach uses a Certification authority, a distributed authorisation model, security models and a secure channel based on SSL [RFC2246]. Their distributed authorisation uses RMI and their secure channel is RMI over SSL. There are no details concerning the Certificate authority. Their distributed authorisation model uses the secure delegation model of Nagaratnam and Lea [NAG98]. It offers three modes for propagating the delegation from a sender to a receiver to a second receiver: the receiver determines the authorisation; the sender determines the authorisation or the sender and receiver authorisation is combined.

Foner [FON96] reports the security model adopted by Yenta. Yenta consists of a set of interrelated mediator agents called Yentas. Multiple Yentas are related using the decentralised authentication model

of PGP [ZIM95]. Each Yenta combines PGP authentication of an agent with signed referrals of that agent from other Yentas. To prevent Yenta's code from being hacked, Yenta source code[1] is signed.

HU [HU01] has reported use of PKI for authentication and SPKI (Simple PKI) model for authority delegation in FIPA MASs. Two kinds of identity certificates are used: one for agents and one for agent owners.

2.5 Current Status of FIPA MAS Security: Summary

At this time (2002), FIPA does not have a strong agent security model mainly because FIPA felt that the issue of where and how to add security to FIPA ACL-based systems needs much more debate. Hence, the current FIPA specifications cannot address the security requirements required for the use-cases described in section 1.

However, just because security isn't specified at the ACL level, this doesn't mean that agents can't have security – security can be accessed at a non-ACL level (see previous section). The main issue is that if FIPA does not specify how to standardise MAS security, interoperability between heterogeneous FIPA MAS applications that require security will be much more difficult.

In the next section, some of the important design issues in specifying MAS security are discussed.

[1] This option is only useful if the source-code is distributed, otherwise the object code can be hacked.

3 FIPA MAS security – 2002 onwards

The previous section has summarised the current state of MAS and FIPA-based MAS security as of Spring 2002. In this section, we review the barriers and potential for specifying and using MAS security. We also discuss some design issues for MAS security and propose some directions for future agent security activities for FIPA.

3.1 Some Barriers to specifying and using MAS security

As we have indicated in previous sections, agent systems can have security without security being modelled at the ACL level. The potential barriers to specifying MAS security at the ACL level include:

1. Security is complex; agent systems are just specialised distributed systems. Secure distributed systems can only be developed by, or have already been modelled by, non-agent security experts - delegate security development issues to them;

2. Security is part of the software infrastructure in which the agent platform is embedded and it is outside the scope of an agent architecture;

3. There is no benefit for security to be either monitored or controlled at the level of agents (i.e., at the application layer);

4. Some agent systems do not need security. The early focus on the MAS community was on collaborative, rational, agent services within closed Intranets;

5. Security is domain and platform (implementation) specific - there is no general agent security architecture that is suitable for all applications and implementations. Hence, there is no reason to suppose that a general standard can be specified;

6. Complete specifications and models of agent systems and agent security are needed before we can start to design and build secure agent systems.

These hypotheses are interrelated. At one level, the first four hypotheses all boil down to the belief that security can be handled properly in the supporting infrastructure for agents rather than at the agent layer. To refute or support these four hypotheses we need to understand the similarities and differences between agent-systems and the application domains in which they are being deployed, and conventional distributed systems and their associated domains.

The fifth and sixth hypotheses suggest a clean slate approach coupled with a deductive or top-down approach to developing secure agent systems. This may be useful but there are many agent systems already in operation. These could benefit from bottom-up approaches coupled to inductive development of agent security models.

Now that we have identified some challenges for developing secure MAS that perhaps distinguishes them from traditional distributed systems, let us briefly consider whether or not we can deploy traditional hard security techniques to protect these types of agent interaction. We will informally argue the case that this is not as straightforward as it initially seems, particularly if agents and agencies operate in a dynamic and open service environment.

Let us consider the use of mediator agents. We could for example authenticate write access to the mediator agents - this would help guard against one agent masquerading as another. We could prevent this by introducing authentication for read access but this would interfere with bootstrapping the system and hinder unknown agents from having the option of browsing an agency's information before joining. If the mediator behaves as an intermediary between a first-party and second-party agent, we may need different encryption schemes and privacy schemes for the first party to share information with the second party but to protect it from the third party and vice versa.

For the case of multi-MAS interaction, we can easily protect each agency by using firewalls, secure channels, access control and authentication against agents outside the agency. However, if multiple MAS need to co-operate, we need to reveal some aspects of our system security to other agencies that are autonomous to us. This is complicated because we may be uncertain of how to trust these agencies.

There are several ways in which agents can enhance security:

- Security, risk and trust can be time-variant. Systems need to be able to analyse (to reason about) and adaptively control these. Reactive, proactive and learning behaviours are properties

that agents can have. Agents can adapt security to the underlying communication infrastructure or dynamic application context.

- MASs are decentralised systems consisting of autonomous units that can act cooperatively and competitively. These can function like a distributed web of independent monitors of the system. Depending on the degree of autonomy that is available, agents could independently monitor and configure their own security based on their application and environment contexts [BOU00].
- MASs can be used to integrate various heterogeneous software systems. For example, Q. He [HE98] proposes that MAS be used to manage heterogeneous types of public key certificate chains.
- MAS agents can use negotiation [HU98], facilitation, brokerage and match-maker services to agree upon levels of encryption for secure channels [ZHA01] and interchangeable authentication credentials.
- It is sometimes useful to deliberate about security, for example, credit-card companies can monitor and analyse usage patterns to detect card misuse. Kakker [KAK00] has investigated using agents to reason about the secrecy of passwords to protect routers and to reason about router integrity.
- Multiple agents can cooperate to enhance security. Authentication can make use of agents' persistence over time and refer to past interactions within a wider body of experience pooled from multiple agent interactions, similar to the way people do. Agents can interact with agents to assess the behaviour of a particular agent, or even the infrastructure itself, to build up a distributed model of trust.

3.2 Security, Trust and Privacy Research and Development

The following areas are suggested as future research area for FIPA:

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

3.2.1 Specifying multiple levels of security and the use of adaptable security

It is anticipated that graded, adaptive and re-configurable levels of security will be needed, based on the different services or application domains and their requirements. Therefore one would have to define different groups of mechanisms that would be used in given situations. Some examples of different grades of security requirements could include:

- The choice between *public but integrity verifiable messages* (i.e. readable by all but with certainty that they have not been tampered with), versus *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).

Tables 3 and 4 below, define six main groups of security concepts and threats required in MAS architectures (platform, directory services, transport services, communication language, application services, software mobility). A description of the specific security mechanisms used to implement the proposed safeguards is also provided.

MAS functionality	Functionality description	Threats
Platform infrastructure	Bootstrapping: launching an agent platform, platform services, agent services and applications	Preventing the launch of new services.
		Preventing the launch of new agents or masquerading through fake registration of agents and services.
		Preventing the launch of a platform.
Directory name service	Standard FIPA AP agent services (White and Yellow page directories).	DoS through overloading of the FIPA facilitating agents (AMS, DF)
Message transport service	Enable flexible communication of information	DoS through bandwidth saturation or corruption of transmitted data. Eavesdropping through cryptanalysis.
Communication language	Syntax: Accepted parsing structure for communication	Interfering with the correct operation of agents through sending messages that cause syntax errors.
	Content parsing: Automated parsing of objects for sharing within a message	Fake content that passes the parser but damages the communicational agent itself.
		Sending very large content.
	Semantic models: For sharing and filtering information	Creating semantic information that makes the agent behave in a way that compromises its goals or services.
	Interaction dialogues: Dialogue automation	Automating continuous dialogues that prevent the computational agent from doing any real service delivery (DoS through spamming).
Application specific agent service access	Provides a mechanism for sharing and interacting with actual services within a single domain.	Faking a service request or service delivery.
	Multi-domain service access.	
		Corruption of trust certificate in one domain to gain certification in others.
Software mobility	Movement and sharing of components, agents and context.	Attacks by malicious host on an agent such as rendering a service unavailable, inspecting and changing data, information or strategies, through the use of viruses that make agents operate in unwanted ways. Attacks by agents onto hosts such as DoS attacks on the hosts by resource consumption; eavesdropping, circumventing access controls etc

Table 3. MAS functions, Threats, safeguards and mechanisms

MAS functionality	Safeguards	Mechanisms
Platform infrastructure	Certified agent naming, checked before registration in AMS and DF.	Encrypted agent identity (through unique NONCE, or "genetic information" that certifies the agent's origin. Certification through a CA (e.g. using PKI).
	N/A	N/A
Directory name service	Verification of requesting agent, restriction of service advertisement to authenticated agents. Restricted use of directory until agent becomes trustworthy.	Use of authentication (e.g. digital signatures, encrypted tokens, etc.) before initiation of any request processing. Gradual upgrade of agent privileges.
Message transport service	Restricting use of network resources (through checking of packets at network entry points) based on agent identification.	Authentication of agent. Granting privileges for use of resources through the use of encrypted tokens and timestamps. Communication through encrypted channels by exchanging and using short-life symmetric session keys.
Communication language	Certification of origin, shadow computations that verify the contents acceptability.	Authentication of originating agent, simulation of received message in a protected environment (Sandbox-type security).
	Utilise a speculation component to run the request on a semantic conformance test bed that checks the behaviours and expectations of the agent.	N/A
	Verification of goals and strategies and appraisal of whether requests contribute towards the completion of these goals.	Authentication. Restriction of spamming through a limited number of requests allowed within a time limit.
Application specific agent service access	Verifiable policies of services and agreements. Non-repudiation of messages.	Use of NONCE or encrypted tokens to confirm next stage of transaction.
		Digital signatures for proving knowledge of information and trust.
	Use of recommendations from trusted sources or other domain status certification.	
Software mobility	"Virus protection", certification of source code origin, message integrity.	Simulation of received code in a protected environment (Sandbox), HASH of code to ensure that it has not been altered, authentication of originating agent.

Table 4. MAS safeguards and mechanisms

The security threats are variable depending upon the requirements and the environment (see Tables 1). We therefore propose a multi-level of security, for example:

- Level A: systems that do not have the minimum level of security;
- Level B: minimum level of security for MASs that adhere to the Agent Management System specifications;
- Level C: level of security for MASs suitable for agents used for e-business information exchange.

Here are the general requirements for level B: minimal MAS security

- Authentication;
- Message Privacy;
- Detect if Message integrity is breached;
- Access control to key services.

In order to ground this we apply this to the requirements for FIPA Multiple-agent systems that adhere to the FIPA agent management specifications, for example

- Use of signed credentials for authentication;
- Private Key Credentials stored in a secure store;
- Public keys are bound to names using X.509 v3 certificates encoded as a PKCS12 file format in a secure store;
- Authenticate all agents for write access to AMS using digital signatures cross-check against public key credentials.
- Messages can be selectively encrypted under the control of the agents

It is a future activity of the security WG to specify how to support variable security threats and requirements. Security policies are likely to play a key role here.

3.3 Policies

The concept of policies is to explicitly define the type of conditions a particular set of computational services will adhere to when operating in a particular team. This approach provides more openness to the service architecture as the computational service must 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 and the matching of policies can be done through a set of meta-constraint satisfaction rules.

Some example security policy could include:

- Access Policy that defines access rights and privileges to protect assets from loss or disclosure by specifying acceptable use guidelines for users, operations staff, and management. It should provide guidelines for external connections, data communications, connecting devices to a network, and adding new software to systems;
- Authentication Policy that establishes trust through an effective password or public key policy, and by setting guidelines for remote location authentication and the use of authentication devices (e.g., one-time passwords and the devices that generate them);
- Safeguard choice policy: specifies the required, or preferred, security safeguards. These may be mapped using some covering model to a set of threat and threat effects;
- Accountability Policy or Appropriate Use Policy (AUP): defines the responsibilities of users, operations staff, and management. It should specify an audit capability and provide incident handling guidelines, i.e., what to do and who to contact if a possible intrusion is detected;
- Privacy Policy: defines reasonable expectations of privacy and user legislative rights regarding such issues as monitoring of user processes such as electronic mail, logging of keystrokes, and access to users' files.

It is anticipated that if such types of security domains and policies are modelled using ontologies, multiple security ontologies will be needed.

3.4 ACL security

When considering the impact of security on agent communication paradigms within an agent system, we need to consider at what layer or layers of the communication infrastructure security should be accessible. If we consider the ACL as a set of four layers: transport level, speech-act or communicative act level, ontology level and interaction protocol level, we examine what issues should be considered with respect to providing security at each of these levels, and what the effects of providing security at each of these levels might be.

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

The question arises as to how much of the message it is appropriate to secure. For instance, it may be necessary to secure content differently from the "speech act" header, so that the agent infrastructure cannot spy on the content of the message.

If security interactions are fundamentally different from those covered by the current set of FIPA interaction protocols, then the development of new interaction protocols may also require the specification of new speech acts. Another motivation for proposing new speech acts would be that there is a fundamentally different semantics for the message, and this different semantics should not be buried in the message content. This would happen, for instance, if the infrastructure needed to interpret the message in a special way.

Messages to middleware agents, such as those that accept proxy and propagate CAs (Communicative Acts) may contain nested content that is forwarded to another agent without being accessible to the middleware agent itself. This suggests that the ACL and content languages need a way to assign different security attributes to various (nested) parts of an expression. As information is relayed through communication links with different degrees of security, it may be necessary to encrypt different parts of messages at different times. For example, a message from a hospital to a general practioner may contain private information about a patient. If the hospital-GP link is secure then the message content need not be encrypted. However, if the GP then relays the information to the patient over an insecure link, then the content should be encrypted even if the outer ACL message is not.

Even within the outer ACL layer of the message, there may be a need to have different security attributes for different message parameters. For example, a matchmaking agent may provide an anonymous communication channel between two agents by forwarding their messages to each other with the sender field encrypted.

Note, however, that if message security were to become reified at the ACL layer rather than an all-or-none function of the message transport layer, and different parts of a message were allowed different security attributes, this would complicate the definition of message encodings.

3.4.1 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 into 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 it may appear to the agent. In addition, very low-level network layer encryption is not likely to be end-to–end.

3.4.2 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, [HE98a] 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 FIPA agent management specification but rather than introduce new speech acts, an ontological approach was adopted. 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.

Foner [FON96] was one of the first agent researchers to discuss the problem that many 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.

3.4.3 Ontology level

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

3.4.4 Interaction Protocol Level

One key argument for providing security 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).

An example Interaction protocols for authentication

1. Agent A makes a request to the AMS agent B to offer a service on its agent platform;
2. Agent B checks its policy for allowing access;
3. Agent B knows the name of A from the message and asks A for its credentials;
4. Agent B checks the credentials for A (check identity, check authority);
5. Agent B informs A that its access is permitted or refuses it;
6. Agent B provides Agent A with credentials to offer a service (the policy is that service providers must have credentials to offer services).

The use of interaction protocols may make the security requirements more complex in order to guard against replay attacks and chosen-text attacks being used during the interaction. In addition, the need for secure interaction sequences mean that the envelope security models are insufficient to protect message sequences, unless the message envelope security model includes constructs or handles for message sequences and these constructs can be secured.

3.5 Trust, security and privacy

It is relatively well understood how to secure a closed multi-agent system, where access can be controlled and restricted. Security is of great importance when dealing with open systems in wild environments, such as the Internet, where the objective is potentially to let any new party dynamically join. Security of multi-agent systems is the first requirement prior to any commercial deployment of agent services. Some of threats are common to any information system on the Internet, e.g., eavesdropping, traffic-analysis, masquerading and denial of service. For these, more or less efficient safeguards have already been found. The real problem comes from new threats, which are linked to the proper characteristics of the agents:

- *Privacy*: personal agents encapsulate some personal information about its users, which it must not publicize to any other agent. Moreover, when required, communication between agent should rely on some agreed level of confidentiality;
- *Trust*: in a dynamic environment, parties involved in a co-operation might not have prior knowledge on each other. In order to work efficiently, these parties need to know the level of confidence they can have in the fact that the other party is actually what it claims to be and also in the fact that the other party can actually do what it proposes to do. This requires some kind of standardized authentication or certification mechanism.

In open electronic service environments, the deployment architectures will need to work with the classic concepts of security, enabling social concepts of communication to be captured within the model of interaction and to deal with legal aspects of privacy. The legal requirements may be easier to deal with if privacy agents can operate under maximum attainable pseudonymity.

Security concerns itself more with the protection of the information and processing of the action rather than the entrusting of actions or information. Trust models can complement security models in order to capture the risks in protecting systems and perhaps to propose strategies for dealing with high-risk situations. However, trust is still very much a research issue as there is a lack of consensus on the definition of trust, of well-defined mechanisms to implement trust and of how to combine trust with security.

4 Recommendations and Conclusions

This white paper has reviewed the current agent security concerns, illustrating these main issues using simple case studies. The security concepts in the current FIPA specifications have been examined. Some directions for future work in order to specify security for open agent service architectures have been proposed.

At this time (2002), FIPA does not have a strong agent security model mainly because FIPA felt that the issue of where and how to add security to FIPA ACL-based systems needed much more debate. Just because security isn't specified at the ACL level: this doesn't mean that agents can't have security – security can be accessed at a non-ACL level.

This white-paper recommends that the first priority should be to review and if necessary to modify the following specifications with respect to agent security:

- [FIPA00001] FIPA Abstract Architecture Specification;
- [FIPA00023] FIPA Agent Management Specification;
- [FIPA00067] FIPA Agent Message Transport Service Specification to review the use of the encrypted field in the envelope.

Once the existing specifications have been reviewed, the following areas are suggested as future areas for FIPA specifications:

- Modelling and specifying multiple levels of security and the use of adaptable security;
- Trust, Privacy and security issues and policies;
- Modelling and specifying security at the ACL level;
- Architectural Abstractions, services and designs for MAS security.

If services such as negotiation, personalised access and local context awareness are to be supported by agent technology then security becomes necessary to support: the legal concerns for data protection and the use of personal preferences; social and moral concerns and to promote the general user acceptance by the business community.

5 Acknowledgements

The views in this document represent the views of the FIPA Security WorkGroup (WG) at this time. The Security WG thanks those that have contributed to this white-paper and to reviewing this white-paper. It thanks the agent community for responding to its Request for Information – this has been used as input to this white-paper. In addition, the FIPA security WG acknowledges the input of the FIPA membership and wider agent community during the FIPA meetings and the input from the FIPA security email list.

6 References

[AJP] Abrams, M Jajodia S, and Podell H, eds, Information Security - An Integrated Collection of Essays, IEEE Computer Society Press (January 1995).

[BOU00] Boudaoud K, Labiod H, Guessoum Z and Boutaba R. Network Security Management with Intelligent Agents. Proc. IEEE/IFIP Network Operations and Management Symposium, Honolulu, Hawaii, (April 2000).

[DEC] Decker, K., Sycara K. and Williamson M. Middle-agents for the Internet, Proc. 15th Int. Joint Conf. on Artificial Intelligence, Nagoya Japan, (1997) 578-583.

[DRA] Drake DL, Morse K L. Analyzing the Byzantine Princes with Trust Models. Proc. 4th Int. Conf on Electronic Commerce Research, ICECR-4, (2001) 360-381.

[DIM01] Dimitrakos T and Bicarregui J. Managing Trust in e- Services. Proc. 4th Int. Conf on Electronic Commerce Research, ICECR-4, (2001) 390-396.

[FIPA00001] FIPA Abstract Architecture Specification., Version J, http://www.fipa.org/repository.

[FIPA00023] FIPA Agent Management Specification. http://www.fipa.org/repository

[FIPA00029] FIPA Contract Net Interaction Protocol Specification. http://www.fipa.org/repository

[FIPA00033] FIPA Brokering Interaction Protocol Specification. . http://www.fipa.org/repository

[FIPA00067] FIPA Agent Message Transport Service Specification. http://www.fipa.org/repository.

[FIPA00080] FIPA Personal Travel Assistance Specification. http://www.fipa.org/repository.

[FON96] Foner LN. A security architecture for multi-agent match-making. Proc. ICMAS (1996).

[GHA01] 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).

[HE98a] He Q, Sychara K, Finin T. Personal Security Agent. KQML-based PKI. Proceedings of (AA'98) Autonomous Agents (1998).

[HE98b] He Q, Sycara KP. Towards A Secure Agent Society. ACM AA'98 Workshop on "Deception, Fraud and Trust in Agent Societies", (1998).

[HU01] Hu Y-J. Some thoughts on Agent Trust And delegation. Proc. 5th Int. Conf. on Autonomous Agents, AA2000, Montreal, (2000) 489-496.

[JAN99] Jansen W and Karygiannis T. Mobile Agent Security, National Institute of Standards and Technology Special Publication 800-19 (August 1999)

[KAK00] Kakkar P, Carl Gunter A, and Abadi M. Reasoning About Secrecy for Active Networks. Proc. Comp. Security Foundations Workshop, 2000.

[NAG98] Nagaratnam N and Lea D. Role-Based Protection and Delegation for Mobile Object Environments. ECOOP'98 Workshop on Secure Internet Mobile Computations, Brussels, Belgium, (1998).

[NAS96] 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).

[NSA] Greg Stocksdale. NSA Glossary of Terms Used in Security and Intrusion Detection. http://www.sans.org/newlook/resources/index.htm.

[OC00020] FIPA 98 Part 10 Version 1.0: Agent Security Management Specification (obsolete). http://www.fipa.org/repository/obsoletespecs.html

[PISA] Privacy Incorporate Software Agent (PISA) project. http://www.tno.nl/instit/fel/pisa.

[POG01] Poggi A, Rimassa G and Tomaiuolo M. Multi-User and Security Support for Multi-Agent Systems. Proc. of WOA 2001 Workshop, Modena, (Sep 2001).

[RFC822] Crocker D.H (ed.). Standard for the format of ARPA Internet Text Messages. IETF Request for Comments 822. www.ietf.org/rfc/rfc822.txt (1982).

[RFC2246] Dierks T and Allen C. The TLS Protocol, Version 1.0. IETF Request for Comments 2246. www.ietf.org/rfc/rfc2246.txt (1999).

[RFC2630] Ramsdell B. (Ed.). S/MIME Version 3 Message Specification IETF Request for Comments 2630. www.ietf.org/rfc/rfc2630.txt (1999).

[TNI] Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria, NCSC-TG-005, National Computer Security Center, (July 1987).

[ZHA01] Zhang M, Karmouch A and Impey R. Towards a Secure Agent Platform based on FIPA. Proc. MATA 2001. Springer-Verlag. LCNS, (2001), Vol. 2164, 277-289.

[ZIM95] Zimmermann P. The Official PGP User's Guide, MIT Press, (1995).

7 Appendix A: Security Glossary

Authentication: (1) To establish the validity of a claimed identity. (2) To provide protection against fraudulent transactions by establishing the validity of a message, station, individual, or originator. (3) Means of verifying an entity's (for example, individual user's, machine's, or software component's) eligibility to receive specific categories of information. [AJP]

Authorization: Access rights granted to a user, program, or process. [AJP]

Capability: A protected identifier that both identifies the object and specifies the access rights to be allowed to the accessor who possesses the capability. In a capability-based system, access to protected objects such as files is granted if the would-be accessor possesses a capability for the object. [AJP]

Certification Authority (CA): an entity in a public key infrastructure (PKI) that can certify keys by signing them. Usually CAs form a hierarchy. The top of this hierarchy is called the root CA.

Confidentiality: (1) The assurance that information is not disclosed to inappropriate entities or processes. (2) The property that information is not made available or disclosed to unauthorized entities. (3) The prevention of the unauthorized disclosure of information. (4) The concept of holding sensitive data in confidence, limited to an appropriate set of individuals or organizations. [AJP]

Cryptography: (1) The principles, means, and methods for rendering information unintelligible, and for restoring encrypted information to intelligible form. (2) The transformation of ordinary text, or "plaintext," into coded form by encryption and the transformation of coded text into plaintext by decryption. Cryptography can be used to support digital signature, key management or exchange, and communications privacy. [AJP]

Data integrity: (1) The property that data has not been altered or destroyed in an unauthorized manner. (2) The state that exists when computerized data is the same as that in the source documents and has not been exposed to accidental or malicious alteration or destruction. [AJP]

Denial of Service (DoS): (1) The prevention of authorized access to system assets or services or the delaying of time-critical operations. (2) Any action or series of actions that prevents any part of a system from functioning in accordance with its intended purpose. This includes any action that causes unauthorized destruction, modification, or delay of service. (Synonymous with interdiction.) [AJP]

Digital signature: A cryptographic method, provided by public key cryptography, used by a message's recipient and any third party to verify the identity of the message's sender. It can also be used to verify the authenticity of the message. A sender creates a digital signature or a message by transforming the message with his or her private key. A recipient, using the sender's public key, verifies the digital signature by applying a corresponding transformation to the message and the signature. [AJP]

Non-Repudiation: Method by which the sender of data is provided with proof of delivery and the recipient is assured of the sender's identity, so that neither can later deny having processed the data [AJP]

Privacy - (1) the ability of an individual or organization to control the collection, storage, sharing, and dissemination of personal and organizational information. (2) The right to insist on adequate security of, and to define authorized users of, information or systems. Note: The concept of privacy cannot be very precise and its use should be avoided in specifications except as a means to require security, because privacy relates to "rights" that depend on legislation [AJP]

Policy – see Security Policy

Public key cryptography: Cryptography using two matched keys (or asymmetric cryptography) in which a single private key is not shared by a pair of users. Instead, users have their own key pairs. Each key pair consists of a matched private and public key. Public key cryptography can perform (1) digital signature, (2) secure transmission or exchange of secret keys, and/or (3) encryption and decryption. Examples of public key cryptography are DSS (Digital Signature Standard) and RSA (Rivest, Shamir, and Adleman). [AJP]

Public Key Infrastructure (PKI): Public Key Infrastructure, the things an organisation or community needs to set up in order to make public key cryptography technology a standard part of their operating

procedures. There are several PKI products on the market. Typically they use a hierarchy of Certification Authorities (CAs).

Security policy: The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information. [TNI]

[Social] Trust: of a party A in a party B for a service X is the measurable belief of A in B behaving dependably for a specified period within a specified context in relation to X [DIM]

Trusted Computer System - A system that employs sufficient hardware and software integrity measures to allow its use for processing simultaneously a range of sensitive or classified information. [TNI]