Towards improved trust and security in FIPA agent platforms

Stefan Poslad

Imperial College of Science, Technology and Medicine, Exhibition Road, London, SW7 2BZ, UK.
Tel. +44 171 5946319
Email: s.poslad@ic.ac.uk

Monique Calisti

Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland.
Tel. +41 21 6936677
Email: calisti@lia.di.epfl.ch

ABSTRACT

FIPA (Foundation for Intelligent Physical Agents) specifications are being proposed as the open standards for heterogeneous agency interaction. We examine the notions of trust and security inherent in the core or normative FIPA specifications and in the existing preliminary security specifications. We highlight its strengths and weaknesses and discuss the steps needed to improve security in the FIPA agencies.

Keywords

Multi-agent system security, agent standards, trust, facilitator agents.

1. INTRODUCTION

Multi-agents (MA) systems are types of distributed systems which consist of autonomous entities called agents, and which routinely use rich social interaction with other agents to complete tasks they have been assigned. Early agent systems consisted of proprietary closed environments in which collaborative and largely benevolent agents operated. Today's agents are being routinely deployed in open distributed networks such as the Internet. In order for agents to be deployed in domains such as e-commerce in such an open infrastructure, agents cannot assume that they are interacting in environments which are inherently safe, and with entities which are benevolent. Increasingly agents require confidentiality, integrity, availability, accounting and non-repudiation in order to be of service.

In addition, there are many proprietary agent systems whose agents use their own non-standard languages and protocols to communicate, these are unable to communicate with others in other heterogeneous agent systems. Thus, not only is security important for communication but increasingly communication standards are becoming important for agents too. The dominant de facto standards for communicative agents are the Foundation for Intelligent Physical Agents (FIPA) specifications. There is a growing number of agent projects, platforms and agent applications which are based on the FIPA standard [1].

We do not aim to reiterate the general analysis and classifications of attacks and possible countermeasures for securing agent technology described as part of published agent systems which address security such as [2],[3] and [4]. We also do not address security requirements and designs for mobile agents - the focus for the majority of papers on agent security such as [5],[6],[7].

Instead, in this paper, we focus on the type of static, possibly intelligent, autonomous agents, which communicate using an Agent Communication Language or ACL. We analyze the types of security guarantees and trust models defined in FIPA specifications for communicative type agents. In this analysis, we also consider whether or not there are security needs that are common to all agent-oriented applications, i.e., that are intrinsic to the agents' nature and the environments in which they operate.

This paper is organized as follows: in section 2, we describe the FIPA agent model in general. In section 3, we analyze and describe the current FIPA security and trust models. Section 4 discusses some general design issues. Section 5 contains the summary and conclusions.

2. FIPA AGENCIES

FIPA is a non-profit standards organization established in 1996 and registered in Geneva, Switzerland [8]. Its purpose is to promote the development of specifications of generic agent technologies that maximize interoperability within and across agent based applications. There are two normative or core (sets) of FIPA specification: those dealing with the agency and those dealing with the Agent Communication Language. We discuss each of these in turn.

FIPA agents operate within the context of FIPA agencies (called FIPA agent platforms) which provide typical generic support services, also called middleware services. FIPA agencies manage their life-cycles, enable them to provide and access services and supports communication with other agents in the same and in different agencies. Current agent middleware services, except for the communication services, are provided by two core FIPA middle agents (also called facilitator agents) called the Agent Management System (AMS) and the Directory Facilitator (DF) respectively. With the exception of the life-cycle management roles, these agents behave similarly to the Retsina MA system agents described in [1].

All service user agents and service provider agents must register themselves with the AMS which offers a white-page service. This registration process allows the definition of a ‘contract’ between any agent and the AMS in order to enable the AMS to manage their life-cycles.

There is current debate within FIPA as to whether these middleware services ought to continue to be accessed solely through service provider agents. For example, the service for communicating with agents in other agencies was in older specifications referred to as the FIPA ’97 specifications [9] and provided by an agent called an Agent Communication Channel or ACC. This has been criticized as being both inefficient, e.g., an agent must send a forward message to an ACC to get it to send a message to the other agent residing in another agency, and perhaps too unmanageable to integrate or embed within a non-agent service infrastructure. Thus, in the current agency model, the ACC still exists as a communication service but it is no longer an agent, it is invoked via some internal API. All other middleware services including the security services may also follow suit. There are pros and cons to “agentising” middleware services. The main advantages are that as agents speak a universal language, the FIPA ACL, services have a natural rich interface for communication. The main disadvantage seems to be that these middleware services are considered so complex and time-consuming to develop to become efficient, robust and manageable and embeddable within non-agent infrastructures, that existing off-the-shelf-[non-agent] services ought to be reused and accessed via a standard Application Programmer’s Interface.

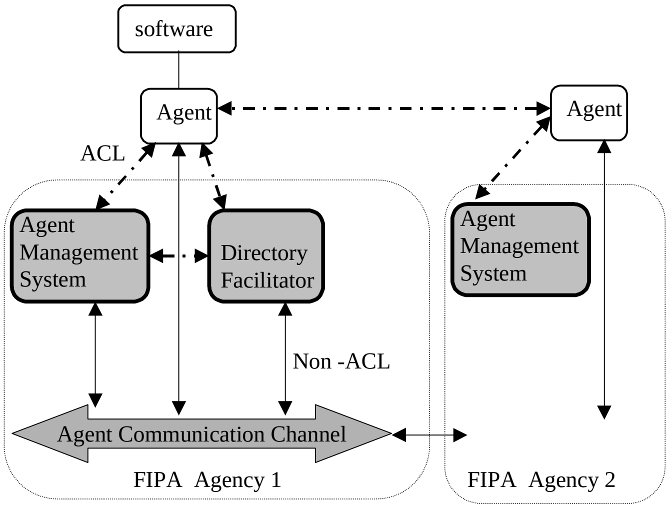


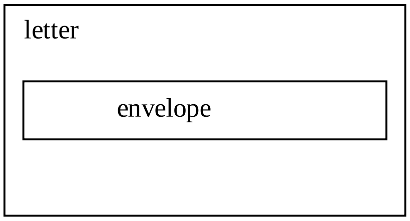
Figure 1. Agent interoperability between two different FIPA agencies

FIPA agents communicate through the use of the FIPA Agent Communication Language or FIPA ACL. The minimal requirements for an agent to be FIPA ACL compliant are:

- Capability to understand and send a not-understood message.
- Capability to correctly implement ACL messages according to their syntactic definition.
- Capability to correctly make use of ACL performatives according to their semantic definition.
- Capability of generating messages in the transport form that corresponds to the messages they want to send.

The first element of an ACL message is the name of the performative. The rest is a sequence of message parameters that can occur in any order in the message. Furthermore, for messages to be processed, e.g., forwarded, to the right recipient/s by the Agent management System, AMS, a letter construct containing the ACL message and the envelope component is built.

Figure 2. The FIPA ACL letter construct



3. CURRENT FIPA SECURITY AND TRUST MODELS

Although, specifications pertaining to security within the context of the FIPA specifications were started at the beginning of 1998, the FIPA 98 agent management specification [10] and the FIPA 98 agent management security specification [11], there is still no coherent, completed picture for agent security within FIPA at this time. In fact both of these specifications have now been declared obsolete by FIPA – the management specification has been superseded by new specification but which contains no references to security. Nevertheless, we use these as a reference point for our discussion on FIPA security below because they represent FIPA’s last published viewpoint on security.

Before we discuss these particular FIPA specifications in detail, it is worth considering why there are no completed current specifications for agent security within FIPA. This is perhaps related to a more general question of whether a generic or default level of agent security ought to be specified that can be applied to different types of agent infrastructures and application domains.

These discourse can be summarized as:

- Security is very complex and secure systems can only be developed by security experts and not by agent system developers.
- Security is part of the software infrastructure in which the agent platform is embedded and is outside the scope of an Agent architecture
- Agents do not need to carry-out a discourse on security configuration at the ACL level
- Security is domain and platform (implementation) specific - there is no general agent security architecture which is suitable for all applications and implementations

- The focus has been the development of collaborative, rational agent services within Intranets – some agents systems don't need.

Let us debate some of these points in more detail. The generic forces for security engineering are different from other types of engineering such as application development. Applications are useful for what they can specifically do. Security products are useful because of what they do not allow to be done. A security strategy in general involves figuring out how to make things not work and then preventing those failures at a reasonable cost. Where possible the developers of secure agent systems should seek to tap into the expertise in the existing fields such as network security.

Many of these other points are interrelated. In certain types of business environment such as business to business exchanges over a private network, within the same enterprise - an adequate level of security may be inherent in the infrastructure for agents that behave rationally.

Whilst, it could be argued in particular examples that security should always be invisible to the agent, there is also an opposing argument for agents in general to be able to monitor and even control the level of the security they require. Agents which are aware of the security infrastructure in which they operate may be able to rationalize the use of encryption for particular transactions between particular parties. There is a high cost in encrypting all messages where it is not necessary and not all parties may support or desire encrypted communication. Agents may also be able to more accurately reason about and report security breaches and system malfunctions.

3.1 FIPA Trust Model

There is a strong notion of dependency in MA systems on facilitator agents. As peer-to-peer interaction is more open and more dynamic and more complex interactions can emerge on-the-fly, agents can not easily discover and maintain knowledge of all agents they need to interact with. Agents in most agent infrastructures expect to locate facilitators and then to locate other agents through facilitators.

3.1.1 Trust between agents and the Agent Management Service.

All agents using or providing services in a FIPA agency must register with the AMS agent. The AMS is trusted to register and maintains the identity of any agent in the agency. As authentication is not defined and mandated, it is difficult to prevent agents masquerading as agents with other identities. A favorable time to switch identities is when a software agent crashes, another agent can then masquerade as it before it recovers.

In registering with the AMS, agents enter into a 'contract' with the AMS to report significant life-cycle changes to the AMS and to allow the AMS to control their life-cycles. Agents are autonomous entities, the AMS may direct an agent to terminate but it may refuse. In this case, depending on how agents are implemented the AMS, may have recourse via an API to directly invoke operations on that agent to terminate it. Otherwise, the AMS could remove that agent's details from the platform, to all essential purposes making it invisible.

3.1.2 Trust between service providers and the directory facilitator

In addition to all agents registering with the AMS, service provider, agents register themselves with the DF. There is no explicit model of trust between agent service providers and the DF agent. Any agent registered with the AMS has the authority to register their services with the DF. There is no support for keeping parts of directories private to support e-commerce applications such as Business-to-Business (B2B) exchanges. The FIPA DF agent reveals service provider details to anyone that asks.

There are implicit quality of service and availability levels associated with the DF service. There is no quota or penalty on the size, complexity or number of descriptions registered. A malicious provider could register a very long description that may significantly use up a DF's finite storage capacity. Similarly a lengthy description may reduce the DF's search efficiency.

Consider what happens when multiple service providers have registered with the DF to provide the same service. There is no policy of fairness to control how the DF distributes requests to multiple providers registered to provide the same service, e.g., the DF may give preferential treatment to the first provider registered to provide a particular service.

There is a simple life-cycle model for services: services cannot be withdrawn unless they have been previously registered. And only the agent, which registered the service description at the outset is subsequently allowed to modify its descriptions.

3.1.3 Trust between service user agents and the Directory Facilitator

FIPA doesn't define how agent service users can define preferences, service users don't generally reveal their preferences to the DF, they may reveal their preferences to service specific providers. This naturally leads to a pull-type service delivery, and tends to exclude the more proactive push type service delivery. Decker et al [12] describes a richer classification of facilitators to support both pull and push types of service access.

3.1.4 Trust between agents and the Agent Communication Channel

Agents delegate message transport to the ACC. The ACC behaves as a broker between agent users and providers, it channels all requests from users to providers. Depending on how the ACC is designed and implemented, the broker model can be a possible bottleneck. "Bandwidth greedy" or malicious service users (or providers) can "soak up" all the resources in resulting in denial of access to the facilitator.

The ACC is also trusted to transmit the messages in a timely manner and to maintain both the integrity of messages transferred and the integrity of message sequences within a conversation or interaction protocol. For example, the FIPA-request interaction protocol defines a successful request as a triple of messages, request, agree and inform. Messages send by the agent in this order, are expected to be received by the receiving agent in this sequence too.

3.2 FIPA Security Model

A FIPA Agent Security Management Model (Figure 3) was defined in a specification first published in 1998 [11]. However, no FIPA based agent systems reported their use of this model. The model 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 next section).

This model enhanced the roles of the AMS and DF agent and introduced an entity called the APSM (Agent Platform Security Manager) to be specifically responsible for maintaining the agency and the infrastructure security policies. The AMS is primarily responsible for authenticating agents within the agency and describes the use of private and public keys for authentication.

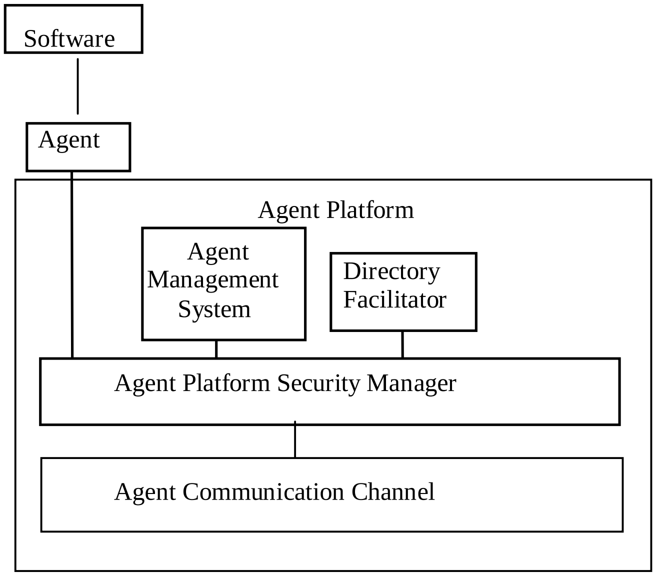


Figure 3. The FIPA agent security management model (this is updated slightly, the ACC was an agent in the original model

Key pairs need to be exchanged between the AMS and agents and stored and used at both the AMS and with the agent owner of the private key – this gives two places for attack instead of one. A second major problem with AMS authentication is its exclusive use of public-private key encryption keys. This requires a complex public key infrastructure containing certificate authorities, which vouch for public and private key holders and this complexity introduces further weaknesses. For this reason, secret key encryption techniques are often used in conjunction with simpler public key infrastructures in place of, public key authentication requiring the use of complex certification authority chains.

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.

3.3 FIPA ACL Security Support

The current FIPA specifications do not provide any specific mechanisms or policy for secure ACL communications. The implicit assumption is that agents are co-operative and trust-worthy and that security is performed elsewhere in the software infrastructure in which the agent is embedded. Agent developers can add security mechanisms at several different levels in a FIPA compliant platform as far as the normative functions (i.e., those functions that must be implemented by an agent system to be FIPA compliant) are still valid. At the communication level some preliminary suggestions for secure ACL communication have been made in the previously described FIPA Security management Specification [11].

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 have specific security `keywords’ such as confidentiality, integrity, authentication and *non-repudiation* by means of which it would be possible to express a level of security or a specific mechanism. The idea is that an agent can request security services, but the responsibility of encapsulating the messages lies with the message transport mechanism.

In the following we show an example of a `secured’ ACL message:

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

Figure 4. An example of a `secure’ ACL message

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.

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, to verify their integrity and to sign them.

4. REQUIREMENTS AND DESIGN ISSUES FOR ADDING SECURITY TO FIPA AGENT SYSTEMS

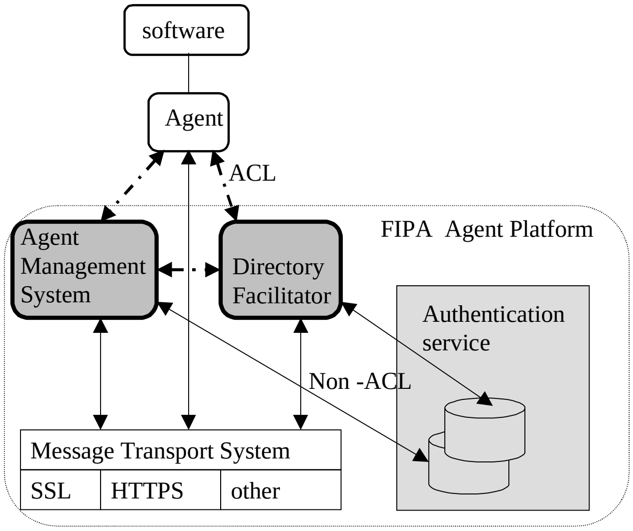


Figure 5. A `secure' agent platform

We propose as a minimum level of security, defenses against attacks by malevolent agents on the agencies and by attacks of malevolent agents on other agents. These defenses include:

- Authentication of agents by facilitators when writing to directories accessed via facilitator agents such as the FIPA AMS and the DF. This helps prevent one agent masquerading as another agent and changing directory information it doesn't own.
- Authentication of facilitators by agents so that agents are able to trust that information and requests sent to them by facilitator agents is valid.
- The use of a private channel for transferring messages between agents when required. This helps prevent malevolent agents stealing private information belonging to others.

The design of a secure agent platform is given in figure 5. Authentication is implemented using a simple public key infrastructure possibly used in conjunction with a private channel such as the Secure Socket Layer. Policies for the distribution, management and certification of keys would need to be defined. The private channel is used for bulk encryption and transfer of messages.

Further work entails:

- Defining the defense roles to be performed by each agent and by dedicated security entities.
- Defining how the security service is exposed at the agent level. This may be simple or complex depending on how much of the security management is accessed and performed at the agent level vs. the software infrastructure level.

5. SUMMARY AND CONCLUSIONS

The need for both standards and security in agent systems has been highlighted. The FIPA specifications have been analyzed with respect to security. Overall the FIPA agency trust models are easy to exploit in order to disrupt access to service providers, to deny services to other users and to masquerade as other users breaking their privacy. The old FIPA Security model is incomplete and has been declared obsolete.

We have outlined some requirements and designs for new security and trust models within FIPA, for use in public networks. This work is ongoing at this time.

6. ACKNOWLEDGMENTS

The author from Imperial College gratefully acknowledges joint support for this work as part of the CASBAh project, jointly funded the UK Engineering and Physical Sciences Research Council (EPSRC) under grant GR/L34440 and Nortel Networks. The views expressed here are those of the authors and do not purport to represent the FIPA membership.

7. REFERENCES

[1] Poslad S. J., Buckle S.J., and Hadingham R. The FIPA-OS agent platform: Open Source for Open Standards. Proceedings of PAAM 2000, Manchester UK, (April 2000).

[2] Wong, H. C. and Sycara K. "Adding security and trust to multi-agent systems", *Proc. Autonomous Agents '99 workshop on deception, fraud and trust in agent societies*, pp 146-161 (1999).

[3] Soshi, M., and Maekawa M. "The Saga security system: a security architecture for open distributed systems", *Proc. 6th IEEE Computer society workshop on future trends of distributed computing systems*, 53-58 (1997).

[4] Thirunavukkarasu, C., Finin T, Mayfield, J. " Secret Agents -- A Security Architecture for the KQML Agent Communication Language", CIKM'95 Intelligent Information Agents Workshop, Baltimore, (December 1995).

[5] Corradi, A., R., Montanari and C., Stefanelli. Security issues in mobile agent technology. Distributed Computing Systems, 1999. *Proc. 7th IEEE Workshop on Future Trends of distributed computing systems*, 3 -8, (1999).

[6] L., Korba. Towards Secure Agent Distribution and Communication, Proc. 32nd Hawaii International Conference on System Sciences, 10pp, (1999).

[7] Farmer, W.M., J. D. Guttman, and V Swarup. Security for mobile agents: Authentication and state appraisal. *Proc. 4th European Symposium on Research in Computer Security*, Springer-Verlag Lecture Notes in Computer Science No. 1146, pages 118-130, (1996).

[8] FIPA Home Page. http://www.fipa.org.

[9] O'Brien, P. D., and R. C. Nicol. "FIPA - towards a standard for software agents", *BT Technology Journal*, Vol.16, no.3, 51-59 (July 1998).

[10] Document OC00019. FIPA 97 Version 2.0 Part 1. Agent Management. Available from http://www.fipa.org.

[11] Document OC00020. FIPA 98 Version 1.0 Part 10. Agent Security. Available from http://www.fipa.org.

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