Agent-Oriented Public Key Infrastructure for Multi-Agent E-Service [⋆]

Yuh-Jong Hu and Chao-Wei Tang

Emerging Network Technology (ENT) Lab. Dept. of Computer Science
National Chengchi University, Taipei, Taiwan 116
{jong, g9017}@cs.nccu.edu.tw

Abstract. Agent is autonomous software that mediates e-service for human on the Internet. The acceptance of agent-mediated e-service (AMES) is very slow for the lacking of security management infrastructure for multi-agent system. Therefore we proposed an agent-oriented public key infrastructure (APKI) for multi-agent e-service. In this APKI, a taxonomy of digital certificates are generated, stored, verified, and revoked to satisfy different access and delegation control purposes. Agent identity certificate was designed for agent's authentication whereas attributed and agent authorization certificates were proposed for agent's authorization and delegation. Using these digital certificates, we establish agent trust relationships on the cyberspace. A trusted agent-mediated e-service scenario will be shown to demonstrate the feasibility of our APKI.

1 Introduction

Web technologies have been developed very fast for the past few years. However human still has to spend a lot of time on searching information that he needs. Therefore agent technology provides the capacity to mitigate this problem. If autonomous agent as human's software delegate that can do all kinds of e-services then it really saves us tremendous amount of time on mundane jobs. We give an execution order and delegate our authority to agent, then agent finish the services for us autonomously.

It is still uncertain whether we can deploy agent technology in large scale on the Web. One of the reasons is that it always takes a considerable of risk for human using agent services on the Web. People are still hesitating at using agents as their legal delegates on the WWW [7]. We are going to address several unsolved issues in this paper before people can really apply agent technology on the Web. First, we must guarantee that each agent was tightly bound with its owner for legal responsibility. In case agent has done something wrong on the cyberspace, we can trace its real owner for liability. Second, even all of transaction service agents were bound to their owners, we still need to establish a trust relationship between these agents so that agent can verify each other's identity and its owner's trustworthiness.

[⋆] This research was supported by Taiwan National Science Council (NSC) under grants No. NSC 90-2213-E-004-008.

We proposed an Agent-oriented Public Key Infrastructure (APKI) to resolve above two issues. Similar to Human PKI (HPKI), this APKI also has Agent Certification Authorities (ACAs) to issue, store, and even revoke agent's certificates [10]. Of course, agent can verify each other's identity certificate under this APKI. In addition, each ACA in APKI is in charge of legal binding service between agent and its owner so that agent's owner is responsible for his agent in the real world. If possible, we even allow human delegate the trust validation authority to ACA so that the APKI might subsume the HPKI functions. We also introduce human attribute certificate and agent authorization certificate for authority initialization and delegation [8].

Finally we briefly demonstrate how to build an APKI on the multi-agent system using FIPA-OS [3]. Our APKI complements existing FIPA standards and frameworks that do not have security mechanism for agents to authenticate and authorize with each other [4].

2 Related Work

There were several PKI architectures proposed and implemented recently, such as X.509 PKI, PGP, SPKI/SDSI [1][10]. Each PKI infrastructure has its own format of certificate, name space, and topology to satisfy the requirements of security criteria. Unfortunately, these PKI infrastructures are only designed for human but not for agent. When the software paradigm is shifting from the monolithic and passive system to the newly cooperative and autonomous agent-based system. We need an APKI that is specifically designed for multi-agent system. There has been some studies on agent security and APKI but they did not provide a binding mechanism between human and agent so that we can find the trust path between two e-service agents for the trustworthiness of an agent [2][5][6][11]. We have a binding mechanism between agent and its owner to guarantee the legal responsibility. We also apply agent core technologies on certificate management operations, such as certificate applying, issuing, storing, and revocation so that we can achieve trusted agent-mediated e-services via digital certificates management under our APKI [9].

3 APKI Architecture

Why we separate APKI and HPKI into two frameworks? Because human lives in the physical world that provides the essential trust foundation on any e-service transaction but agent lives on the cyberspace that relies on human's trust linkage. Certainly we can not enforce any existing Human CA (HCA) to provide agent certificate management services.

3.1 APKI Deign Issues

There are several issues on designing of APKI shown as the followings:

1. The APKI has to provide full capacity to manage all of agent identity certificates on the WWW.
2. The human liability for its agent is based on the certificate binding mechanism between human and agent so that nonrepudiation of agent's owner is ensured.
3. The feasible APKI topology needs to meet the efficient and robust of agent certificate management for tremendous amount of agents.
4. The essential trust path between service provider agent and service request agent needs to be set up to ensure the trustworthiness of the service.
5. The APKI might subsume the HPKI if human grants the necessary trust validation authority to all ACAs on the essential trust path.

3.2 Agent Certification Authority

Agent Certification Authority (ACA) is defined as two agents: Registration Agent (RA) and Management Agent (MA).

- **RA**: RA registers to adjacent upper layer's MA to get its identity certificates except RA on the top of APKI who registers to its MA.
- **MA**: MA provides certificate management operations for adjacent lower layer's agent certificates, including certificate issuing, storing, verification, and revocation. What is the possible APKI topology to serve all of agents on the Web? To resolve above APKI design issues, we propose a strict hierarchy tree APKI topology with three layers of ACA and one layer of application agent (see Fig. 1).

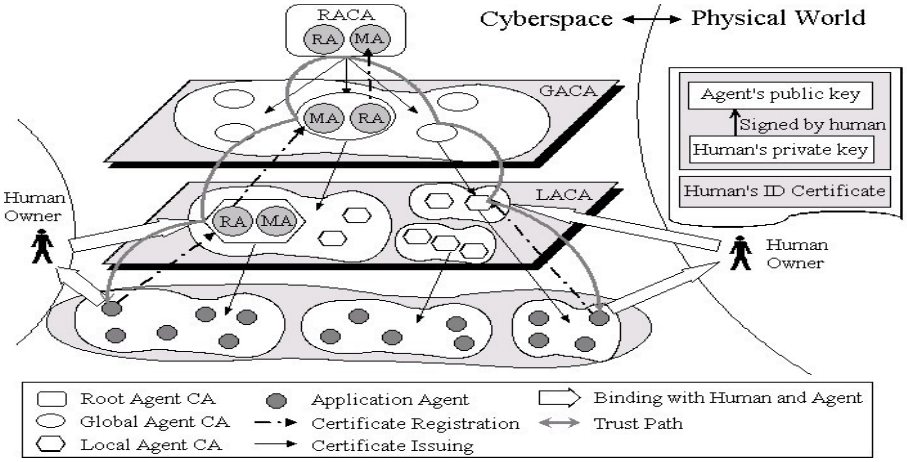

Fig. 1. Agent-Oriented Public Key Infrastructure with three agent digital certificate management layers and one application layer

- **Root ACA (RACA)**: RACA is on the top layer of APKI so every lower ACA must trust it. The RA of RACA is the root of trust so it initially registers to its peer MA to get agent identity certificate.
- **Global ACA (GACA)**: the RA of GACA registers to the MA of the RACA to get its agent identity certificate after the RACA was initialized.
- **Local ACA (LACA)**: the RA of LACA registers to the MA of the GACA to get its agent identity certificate after LACA was initialized.
- **Application Agents (AAs)**: AA is the real service agent for human in our APKI. Before requesting services, AA registers to one of local ACAs it trusts for applying an identity certificate so that other service provider agents could validate this agent's identity and trustworthiness through the trust path.

4 Agent and Human Certificate

4.1 Agent identity certificate

The format of the agent a's identity certificate is shown as the followings:

$$ID_{ACA \mapsto a} - Cert = (Id_a, Pu_a, Sig(Pu_a)_h, V, Option, Sig_{ACA})$$

where Id_a: a's distinguished identity; Pu_a: a's public key; $Sig(Pu_a)_h$: agent a's public key signed by human h's private key, it is human and agent's binding information; V: validation period; *Option*: optional information; Sig_{ACA}: certificate signature signed by ACA's private key.

- **Agent-certificate-applying**: the information required for applying agent certificate are agent's public key, its owner's identity certificate, and agent public key as message digest signed by human's private key to endorse this particular agent (see Fig. 1). We guarantee that human is fully responsible for his own agent so that non-repudiation criteria are satisfied. We did not sign agent's process code as message digest because the process code is mutable.
- **Agent-certificate-issuing**: the MA of ACA issues an agent identity certificate after verifying necessary information, such as owner's authenticity.
- **Agent-certificate-storing**: the MA of ACA stores a copy of agent identity certificate to its repository for other agent's query or validation.
- **Agent-certificate-revocation**: the MA of ACA revokes the agent identity certificate from the repository when this certificate's validation period expires.
- **Agent-certificate-verification**: the establishment of trust path is to ensure the trust relationships of all ACAs in the trust path. Thus ACA is not only endorse the binding relationship between agent and its owner but also propagate the trust beliefs within the path. We could establish the trust path between any two AAs and their associated owners if human delegate the (human) trust verification and propagation authority to the corresponding ACAs.

Human attribute certificate and agent authorization certificate are designed for agent authority initialization and delegation shown as the followings:

4.2 Human attribute certificate

The attribute certificate is only proposed for human. We did not propose agent's attribute certificate because agents do not have initiative authority unless human or institution delegates. The format of attribute certificate is shown as the followings:

$$AT_{TA \mapsto h} - Cert = (Id_h, Ar_h, V, Option, Sig_{HTA})$$

where Id_h: principal h's distinguished identity; Ar_h: principal h's attribute information; V: validation period; $Option$: optional information; Sig_{HTA}: signature signed by HTA's (Human aTtribute Authority) private key.

4.3 Human and agent authorization certificate

The SPKI/SDSI-based authorization certificate was initially created for human to request service from service application agent [1]. This authorization certificate will be re-generated and delegate to agent(s) for further services via different delegation mechanisms, such as chain-ruled, threshold, etc [9]. Even the issuer and subject in the authorization certificate are indicated as anonymous public keys, we still have the capacity to trace who is the original authority delegate and who goes wrong on the delegation process. More specifically we could examine authorization certificates on the chain-ruled and find out the owner responsible for the liability of the agents. The format of human or agent authorization certificate is shown as the followings:

$$AU_{p \mapsto q} - Cert = (Pu_p, Pu_q, A, D, V, Sig_p)$$

where Pu_p: a public key for the issuer of principal p to grant authorization; Pu_q: a public key for the subject of principal q to receive authorization; A: expression for authorization; D: delegation bit with value 0 or 1; V: validation period; Sig_p: certificate signature signed by p's private key.

5 Trusted Agent-Mediated E-Services

A simple agent-mediated e-service (AMES) scenario will be proposed to demonstrate the feasibility of our APKI framework with its associated digital certificates management. In this trusted AMES scenario, we will see that all of the digital certificates we proposed are required for the trust verification.

There exist an e-service portal on the WWW with multi-agent system built on top of it. Agent b on this portal is customized by user (or institution) u_2 as a professional service agent with its specialty, such as ticket reservations. User u_1 might launch his personal agent a to request a service s from agent b.

Initially, assuming that users u_1, u_2 and agents a, b digital identity certificates were issued and stored in the associated HCAs, ACAs in the respective HPKI and the APKI.

1. User u_1 delegates agent a an authorization certificate $AU_{u_1 \mapsto a} - Cert$ for requesting service s from agent b. Of course, u_1 original authority is based on u_1 attribute certificate $AT_{HTA \mapsto u_1} - Cert$.
2. Agent a uses Web Portal's yellow page to locate agent b using b's agent identity certificate. Before an authority delegation was initiated, a trust association will be established between agent b and agent a (shown as step 3). Then a new authorization certificate $AU_{a \mapsto b} - Cert$ will be generated to ask service.
3. A trust association between agent b and agent a is required to ensure the trust criteria satisfaction, including:
   - Agent a and agent b are legally bound with their respective owners. This is an embedded function of LACA.
   - There should exist a trust path between agent b and agent a in the APKI. A trust verification probing message will be sent from the agent b's (or a's) LACA to the connected GACA (or the RACA) to ascertain this condition.
   - Users u_1 and u_2 are mutually trust each other. This criteria are satisfied if all of ACAs in the previous trust path establishment were granted authority from their respective owners.
4. Agent b might serve the agent a's service request directly or it might request service from another agent c. In that case, go to step 2, and a new authorization certificate $AU_{b \mapsto c} - Cert$ will be generated.
5. The complete trusted AMES scenario is achievable if a trust path can be established and all of digital certificates are validated and satisfied with access control rules.

6 APKI Implementation

There are two service management modules in the existing FIPA multi-agent system framework, i.e., Agent Management System (AMS) and Description Facilitator (DF). The objective of AMS is to accept agent's name registration and manage agent's identity whereas the objective of DF is to accept agent's service registration. Agent discovers a specific service from DF with the function of SearchDF.

We built our APKI on the FIPA-OS to evaluate the feasibility of our framework (see Fig. 2). Because there lacks digital certificate management feature in the existing FIPA standards and associated frameworks so our study is very important and significant for the entire agent research community. Adding our APKI module in the FIPA management system certainly enhances agent authentication, authorization, and delegation trust services on the Web.

7 Conclusion and Further Studies

As agent technology is getting popular, we envision the importance of trusted agent e-service using digital certificates. We proposed a tree hierarchy APKI

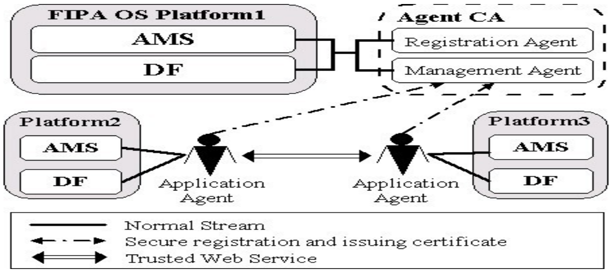

Fig. 2. APKI system implementation framework

topology to manage agent identity certificates. This topology is simple and extensible to serve tremendous amount of agents on the WWW, such as AgentCities. The agent's authentication, authorization, and even delegation are also possible using our taxonomy of digital certificates. We need to establish the trust path between peer to peer agent to secure the e-service trust relation. A trusted agent-mediated e-service scenario was shown and implemented using this APKI to demonstrate the feasibility of our proposed framework. In our further studies, we are using this certificate-based trust establishment theory to build trust ontology and rules on the semantic web.

References

1. Ellison, C. M., SPKI/SDSI Certificates, `http://world.std.com/`
2. Finin, T. W., Mayfield, J., Thirunavukkarasu, C., Secret Agents - A security Architecture for the KQML Agent Communication Language, CIKM'95 Intelligent Information Agents Workshop, Baltimore (1995).
3. FIPA-OS, `http://fipa-os.sourceforge.net.`
4. FIPA Standards, `http://www.fipa.org.`
5. Foner, N. L., A Security Architecture for Multi-Agent Matchmaking, Proceeding of the Second International Conference on Multi-Agent System (1996).
6. He, Q., Sycara, P. K., and Finin, T. W., Personal Security Agent: KQML-Based PKI, Proceedings of the Second International Conference on Autonomous Agents (1998).
7. Heckman, C. and Wobbrock, O. J., Liability for Autonomous Agent Design, Autonomous Agents 98, Minneapolis, MN, USA (1998) 392-399.
8. Hu, Y.-J., Some Thoughts on Agent Trust and Delegation, The Fifth International Conference on Autonomous Agents, Montreal, Canada (2001).
9. Hu, Y.-J., Trusted Agent-Mediated E-Commerce Transaction Services via Digital Certificate Management, Electronic Commerce Research, Vol. 3, Issues 3-4, (2003).
10. Gerck, E., Overview of Certification Systems: X.509, CA, PGP and SKIP, MCG-Meta-Certificate Group, `http://www.mcg.org.br`.
11. Wong, H. C. and Sycara, K, Adding Security and Trust to Multi-Agent Systems, Proceedings of Autonomous Agents '99 Workshop on Deception, Fraud and Trust in Agent Societies, Seattle, Washington (1999).