Some Thoughts on Agent Trust and Delegation

Yuh-Jong Hu
Emerging Network Technology Lab
Department of Computer Science
National Chengchi University
Wen-Shan District, Taipei, Taiwan
jong@cherry.cs.nccu.edu.tw
http://www.cs.nccu.edu.tw/j̃ong

ABSTRACT

In this paper, we are going to show how to build up agent-oriented Public Key Infrastructure(PKI) from SPKI/SDSI and X.509 standards. A variety of delegation mechanisms for agents will be demonstrated under this agent-oriented PKI. The mechanisms include: chain-ruled, threshold, and conditional. The lack of agent security management standards did not allow us to do the agent trust and delegation in legalized manner so we proposed several new communicative acts to satisfy our agent delegation management. Finally, we briefly show how to implement these agent communication language and inner content language in XML and XML/RDF.

1. INTRODUCTION

The agent trust and security issues are very important in the agent-mediated application services, such as e-business [24][25]. If we can not resolve the agent trust and security problems in agent-based research, then it is quite possible that the agent technology can not be applied to the real world applications in legalized manner. Some awareness on agent liability and laws were presented for autonomous agent design in the past autonomous agents conference [16]. They claimed that agent is an end-user autonomous delegate, so the wrong doing of agent might be sued to its end-user or agent system designer. Agent system designer must deliver an agent system that is verifiable so that the system itself will be robust enough to claim its responsibility. So we still consider the most important unsolved problem in agent research is how to build up an agent trust and security framework from security technology perspectives such that we can perform the agent delegation with authentication, authorization, and evidence collections features to enforce those agent liability issues.

There are voluminous of trust definitions, including objective trust, intersubjective trust, and subjective trust [12]. As the relationships between trust and delegation for human vs. agent and agent vs. agent, we regard trust to be a background for delegation purpose. Therefore trust loop is not completely equal to delegation loop. We might trust something(or someone) without granting delegation under some prohibitive conditions so trust is normally *necessary* but not *sufficient* conditions for delegation [4]. In this paper, we did not examine the trust issues before human or agent are granting the authorization certificates to the trustee agents. Instead, our trust and delegation studies are aiming at the trust management mechanisms for services(or resources) guardian agents based on the cumulative authorization certificates from services request agents. These cumulative authorization certificates might be issued directly or indirectly from the services owner's agents or from trusted third party. Thus, the trust issues in our study are concerning about the verification of trustworthiness and validation on agent-oriented Public Key Infrastructure(PKI) with its issuing identity and authorization certificates.

Recently, there were some attempts in solving trust and security for multi-agent systems with conventional security techniques but we doubt their feasibility in the future multi-agent framework design [15][28][29]. Because they did not really solve the agent trust and delegation problem with both authentication and authorization certificates in robust logic manner as in [22][23]. We need a solid foundation for agent trust and delegation via certificate theory. Existing X.509 PKI is based on a very limited trust model making them inadequate for general multi-agent trust management [19]. Furthermore, X.509 PKI is not quite suitable for agent's authority delegation. We need distributed trust management framework, such as SPKI/SDSI, that can be explicitly applied to multi-agent's delegation and authorization in trustworthy manner [6][26].

In this paper, we are going to show how to build up an agent-oriented PKI from SPKI/SDSI and X.509 standards. The SPKI/SDSI attribute-oriented authorization certificate is a token for distributed trust management that can complement the name-oriented X.509 identity certificate. A variety of delegation mechanisms for agents will be demonstrated under this agent-oriented PKI. We also show how agents coordinate with each other to achieve these different delegation mechanisms. Of course, the trust and delegation inference logic will be executed by authenticated verifier agent to verify the correctness of the delegation.

The agent communication language(ACL) with its inner content language for agent trust and delegation will follow the FIPA standards to achieve the maximum interoperabil-

AGENTS'01, May 28-June 1, 2001, Montréal, Quebec, Canada.

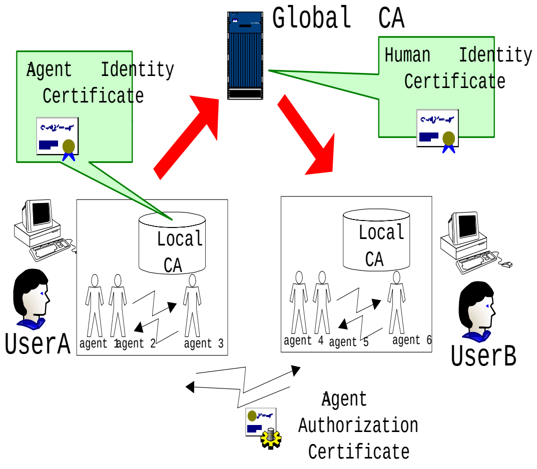

Global CA
Agent Identity Certificate
Human Identity Certificate
Local CA
Local CA
UserA
agent 1
agent 2
agent 3
agent 4
agent 5
agent 6
UserB
Agent Authorization Certificate

Figure 1: The agent-oriented PKI

ity [7]. Unfortunately, the obsolescence of agent security management standards did not allow us to specify the ACL with existing FIPA standard communicative act [8]. Thus we propose some of the new communicative acts that satisfy our agent operation within this agent-oriented PKI.

Finally, we briefly show how these ACL messages will be encoded in XML and their associated inner content language will be encoded in XML/RDF to realize some of trivial advantages proposed by most agent researchers [14].

2. THE AGENT-ORIENTED PKI

A primary function of X.509-based PKI is to bind the public key of an entity(human) to its symbolic name with his associated profile attributes in order to achieve authenticity and non-repudiation [11].

The name-oriented X.509 identity certificate provides the entity authentication services via digital signature. As for entity authorization, the services guardian based on operating system's access control list can differentiate each authenticated entity access rights. Or, we might classify the X.509 identity certificates into several classes to provide different level of authorization. In fact, the X.509 identity certificate does not be easily scaled up for authorization in fine granularity.

We need an authorization certificate, which provides the mechanism to express the agent's authorization and the certificate can be processed in distributed trust manner. So we choose the SPKI/SDSI certificates as our authorization token.

Our agent-oriented PKI must serve both the master(human) and associated slave(agents)(see Figure 1). The global X.509 Certification Authority(CA) provides the identity certificate services for human and local X.509 CAs provide agent's certificate services. As for human/agent vs. agent/agent authorizations, we use SPKI/SDSI certificate to flexibly achieve authorization objectives.

2.1 Identity Certificate

In agent-oriented PKI, the identity certificate is based on the X.509 format with classification as human identity certificate and agent identity certificate.

2.1.1 *Human Identity Certificate*

The human identity certificate $I_h - Cert$ is defined as:

$$I_h - Cert = (ID_h, PU_h, Options)$$

where:
ID_h is human unique symbolic name.
PU_h is unique public key for human.
Options are optional parameters for human profiles, such as email address, birth date, etc.

The purpose of human identity certificate $I_h - Cert$ is to provide the legal liability subject for his agents who are violating the cyberspace laws. The binding of ID_h with PU_h in $I_h - Cert$ allows another human use PU_h to verify the agent identity certificate endorsed by ID_h via his private key PR_h. Furthermore, ID_h might sign his own human authorization certificate $A_h - Cert$ via private key PR_h in the delegation process.

2.1.2 *Agent Identity Certificate*

The agent identity certificate $I_a - Cert$ is defined as:

$$I_a - Cert = (ID_h\#ID_a, PU_a, Options)$$

where:
$ID_h\#ID_a$ is the concatenation of human unique symbolic name ID_h with agent unique symbolic name ID_a.
PU_a is unique public key for agent.
Options are optional parameters for agent profiles, such as agent name, network address , and validity life cycle, etc.

In [5], the local name identity certificate for our agent can be shown as (PU_h, ID_a, PU_a, V) and this agent identity certificate is signed by PR_h to endorse its legal status within its validation period V. The agent identity semantic expression is quite close our notation. Each delegation agent can use his private key PR_a to endorse the agent authorization certificate $A_a - Cert$ in the re-delegation process.

2.1.3 *Global CA vs. Local CA*

The global X.509 CA is installed for human identity certificate operations, which including identity endorsement, registration, query, and revocation, etc. The local X.509 CA is installed in each agent system to provide similar certificate processing operations for agent identity certificate. The public key in agent identity certificate provides the mechanisms for the other agents to verify this identity agent's signature and to encrypt the session key for further message encryption in secret communication channel.

2.2 Authorization Certificate

The human and agent authorization certificate are $A_h - Cert$ and $A_a - Cert$. They are based on the SPKI/SDSI certificate format.

2.2.1 *Human Authorization Certificate*

The human authorization certificate $A_h - Cert$ is shown as a 5-tuple structure:

$$A_h - Cert = (PU_h, PU_a, A, D, V)$$

where:
PU_h is the human's public key for granting authorization.
PU_a is the agent's public key for receiving authorization.

A is the authorization power for agent
D is the delegation bit with 0 or 1 value.
V is the validation period.

2.2.2 Agent Authorization Certificate

The agent authorization certificate $A_a - Cert$ from agent $a1$ to agent $a2$ is similar to the human authorization certificate $A_h - Cert$. The only difference is that both the issuer $a1$ and subject $a2$ for this certificate are agent's public key indicated in the $I_a - Cert$. Thus, the agent authorization certificate can be shown as:

$$A_a - Cert = (PU_{a1}, PU_{a2}, A, D, V)$$

When each agent authorization certificate was issued, this certificate must be signed by issuer agent a_1 private key to ensure its legal status. In order to avoid leaking the issuer agent's private key to the foreign agent system, we insist that the signing of agent authorization certificate must take place in the issuer agent's home system. This constraint excludes the mobile agent category in our study. Even the safe mobile agent signing in foreign host study was proposed in [27], we still consider that the complexity of this results prevent us to release this constraint in the near future.

3. AGENT TRUST AND DELEGATION VIA CERTIFICATES

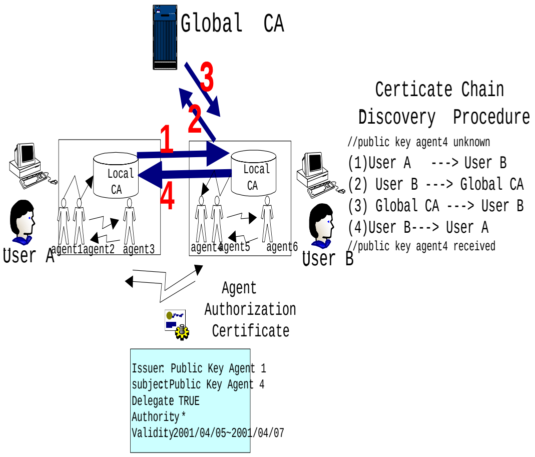
The reasons for people to trust their agents and delegate their authority to these agents are: efficiency, convenience, fault tolerance, and agent is more suitable than human to work in the cyberspace. We regard agent trust and agent delegation to be a primal and dual problem. Trustor human(or agent) might use his(or its) subjective trust criteria with intersubjective and objective trust information to evaluate for trustee agent's degree of trust when proceed the authority delegation. People sometimes differentiate the human social trust from the agent's process trust. In our viewpoint, we consider the human and agent trust domains might cross over so that a unique model of trust might be able to represent both human's social trust and agent's process trust [12].

In this study, we did not evaluate this trust vs. delegation issue. Based on the degree of trust level, if you fully (partially) trust your agent subjectively, then we assume you might delegate your complete (partial) authority to your agent on any Internet-based activities. The complete authority is denoted as speak for and the partial authority is denoted as speak for with role constraint as. So the role constraint allows a principal to delegate his power in a limited partial form.

For example, if the delegation process was initiated at your agent home system, you might subjectively delegate your complete authority to your agent and vice versa. In [4], there are "competence" and "disposition" belief problems shown in the agent authority delegation:

The trustor agent x tries to achieve a goal g via the delegation to trustee agent y

In our certificate-based delegation, the "competence" belief issue did not exist due to the authority certificate itself is a competence token. As long as the authority certificate is issued from trustor agent x to trustee agent y, then the competence token is copied. The "disposition" belief is relative to the *willingness* of the trustee agent y to finish the task. We assume the trustor agent x will evaluate this disposition belief before initiating different type of agent delegation mechanisms, such as chain-ruled, threshold, and conditional, to the trustee agents. In the hierarchical agent organization structure, the subordinate agents always accept the supervisor agent's authority so the disposition belief issue does not exist.



Certicate Chain Discovery Procedure
//public key agent4 unknown
(1)User A ---> User B
(2) User B ---> Global CA
(3) Global CA ---> User B
(4)User B---> User A
//public key agent4 received

Issuer: Public Key Agent 1
subjectPublic Key Agent 4
Delegate TRUE
Authority *
Validity2001/04/05~2001/04/07

Figure 2: Agent trust and delegation via certificates

The agent must be able to fulfill the agent special software characteristics, such as autonomous, pro-active, reactive, and social etc [18]. In [20], similar concepts were applied to the automated message processing in E-Commerce with speech act theory. Agents communicate with each other in speech-act based agent communication language(ACL) with semantic associated with each performative. Besides, each agent is able to parse the ACL content language and use the internal inference mechanisms to change the agent state(mental).

If your agent trust the other people's agents, then your agent might re-delegate your complete (or partial) authority to those agents under your permission. We do believe that there must a robust mechanisms to model and execute both human/agent and agent/agent delegation and consecutive re-delegation process. In such a way, the human/agent and agent/agent trust problems can be solved simultaneously. We think that certification theory with associated trust study results provide a solid foundation to solve both human/agent and agent/agent trust and delegation problems.

3.1 The certificate chain discovery

Agent-oriented PKI must have a mechanism to provide the certificate chain discovery mechanism for any human to discover the peer side human and agent public keys so that we can easily verify the human/agent identity certificate and to do the delegation via human/agent authorization certificate(see Figure 2).

The trust of identity certificate is to trust the legal status of human/agent symbolic name with its public key while the trust of authorization certificate is to believe that the service request agent does have the authority with respect to the presented authorization certificates.

4. DIFFERENT TYPE OF AGENT DELEGATION

The delegation concepts were originally proposed by Lampson et al. in their pioneering papers [1][21]. The idea of their delegation study is to solve the delegation problem in the distributed systems due to the lack of scalability feature in naming and security issues from the centralized systems viewpoints. Later on, some of studies further extends their results for different type of delegation mechanisms with deep delegation logic inference [22][23].

4.1 Chain-Ruled Delegation

In the generic chain-ruled delegation, the authority delegation source is usually responsible for the final authority verification [3]. The delegation mechanism allows the authority be delegated in a cascade style. Therefore, the final service(authority) request agent does not necessarily to be the direct authority delegatee from the service(authority) source.

This delegation type was also able to apply in the safe deals between strangers scenario and when the authority verification agent is different from the authority(or role) assignment agent [13][17]. For example, one university provides digital library online magazine query services only for its staffs and students. The school library computer agent will be responsible for the verification of each access but the role assignment for staffs and students might be done from school another department agents.

4.2 Threshold Delegation

The threshold delegation diversifies agent delegation pattern. If the agent system allows the threshold delegation, then multiple agent delegation subjects are permitted from one agent delegation issuer. Thus, the issuer agent's authority can be split in this single delegation. Incidentally, the delegate agent subjects must coordinate with each other to perform the delegation.

For example, An organization general manager *Bob* has the full authority to do his organization bank account *ABC* transaction on Internet bank *Morgan.* This general manager *Bob* would like to delegate the *check,* and *withdraw* partial authority for bank account operations to 2 of department managers if they both agree to exercise this authority.

The *Bob* identity certificate $I_{Bob} - Cert$ and his agent G identity certificate $I_G - Cert$ are shown as the following:

$$I_{Bob} - Cert = (ID_{Bob}, 123456, Options)$$

$$I_G - Cert = (ID_{Bob} \# ID_G, 783452, Options)$$

Then general manager *Bob* delegates the full authority to his personal agent G [1]:

General Manager *Bob* delegates the operations of *(check, transfer, withdraw, deposit)* to his agent G on Internet Bank *Morgan* with account ABC from 2001/04/05 to 2001/04/07.

This delegation message might be shown as human authorization certificate in the following:

$$A_{Bob} - Cert = (123456, 783452, Au, 1, v)$$

where:
123456 is the Bob's public key.
783452 is the agent G's public key.
Au is the indication for authority expression.
1 indicates the re-delegation bit is positive.
v is validation period for this certificate, i.e. 2001/04/05-2001/04/07.

General manager Bob's public key 123456 was extracted from his human identity certificate $I_{Bob} - Cert$, which are stored in the global X.509 CA. In reality, Bob has his own private key and public key in his local storage. Private key is for authorization certificate endorsement and public key is for the identification purpose. Agent G's public key 783452 was extracted from the agent identity certificate that was stored in local X.509 CA. Au is an authorization tag that indicates the *check, transfer, withdraw,* and *deposit* operations. Conceptually, the agent authorization expression are more high level than file system operations, such as read, write, execute.

Three department mangers create three agents M_1, M_2, M_3 in their respective agent system and the agent G uses threshold delegation to delegate the operations of *check, withdraw* to two of agents in (M_1, M_2, M_3) on Internet bank $Morgan$.

The expression might be shown as:

Agent G says threshold $(2, [M_1, M_2, M_3])$ speak for agent G for the operations *(check, withdraw)* on Internet bank *Morgan* with account ABC from 2001/04/05 to 2001/04/07.

The says and speak for is equivalent to the delegate...to shown as above. The threshold delegation can not be shown explicitly in the SPKI/SDSI 5-tuple certificate but we can show this threshold constraint within authorization field. The SPKI/SDSI-based agent authorization certificate might look like the following:

$$A_{M_1} = (783452, 254416, Au, 1, v)$$

where:
783452 is agent G's public key.
254416 is agent M_1's public key
Au authority was reduced to *check, withdraw* with threshold conditions.

The threshold condition specified in Au field will enforce agent M_1 to initiate coordination process with one of the agent in (M_2, M_3). Similar agent authorization certificates A_{M_2} A_{M_3} were also sent to agent M_2 and agent M_3 with minor modifications for authority subject's public key and threshold conditions.

[1] The general manager might delegate only *check* and *withdraw* partial authority to its own agent G based on its own subjective degree of trust evaluation to his agent G.

4.3 Conditional Delegation

The conditional delegation is specified in the authorization tag Au in the authorization certificate. We might regard the threshold criterion to be one of the conditions. The re-delegation and time validity conditions were specified in separate fields so we did not show these two conditions in Au.

In [21], the re-delegation bit is always set to 1 for permission of re-delegation. In their viewpoint, if re-delegation is set to 0, then any vicious subject agent might copy his private key for another delegatee agent to achieve authority re-delegation purpose. Unless the authority verifier agent do have a policy in its rule base to restrict the depth of re-delegation, we do not have a robust way to set the re-delegation bit to be 0.

The general phrase to express the conditional delegation can be shown as the following:

Human(or Agent) delegates the operations of *(set of operations)* to agent G(or *threshold(m, [A_1,..,A_n])* on an *application* domain under the conditions that *satisfy* some regular conditional constraints within validation period.

The delegates is the authority permission from issuer agent via signing the 5-tuple authorization certificate $A_a - Cert$. If authority delegation are under threshold condition, then this condition will be shown explicitly to enforce these delegatee subject agents to autonomously coordinate with each other. Of course, all of the agents $\in [A_1,..,A_n]$ will receive the same $A_a - Cert$ from the issuer agent. The *operation set* is an indication for full or partial authority from issuer agent. An *application* domain is the domain of delegation that can be shown as shared ontology. Finally, the validation period is a standalone condition that is out of general conditions to indicate the $A_a - Cert$ time validity.

We reiterate the Internet bank example with amount and time conditions shown as the following:

General Manager *Bob* delegates the operations of *(check, transfer, withdraw, deposit)* to his agent G on Internet Bank $Morgan$ with account ABC under the conditions that the amount of withdraw no more than 10000 dollars in one day starting from 2001/04/06 within 2001/04/05 - 2001/04/07 validation period.

The conditions on this example show that the authority might be valid in the future if the authorization certificate was issues on 2001/04/05 and the constraints will be applied to some(or one) of the operations in a certain amount. This authorization conditions are satisfied by the subject agent alone. In case the threshold delegation, the conditions will be applied to all of the subject agents to exercise the authority.

The delegation mechanisms for SPKI/SDSI authorization certificate is more flexible than simple role-based access control delegation shown in [17]. The authority for role-based access control is only specified in resource allocation agent site but the authority for SPKI/SDSI delegation can be dynamically configured in the authorization certificate and coped with different delegation mechanisms.

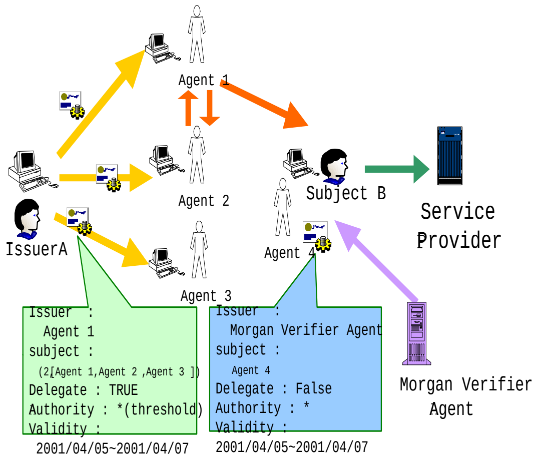


Figure 3: Agent trust and delegation network

5. THE AGENT TRUST AND DELEGATION LOGIC

The trust and authorization problem consists of deciding whether the incoming collected certificates(or credentials) prove that a services(or resources) request complies with a human specified policies. The implementable and tractable delegation logic concepts were proposed by Li et al. in their distributed trust management research [22][23]. In their revised D1LP(Delegation Logic Programs version 1), there is a restriction imposed in this model in order to ensure the inference computation tractable. The constraint is that a delegatee(subject) appearing in a trust policy rule body, or in a query, must be a principal or a conjunction of principals so delegatee is not permitted to contain a disjunction or threshold structure.

Consider the chain-ruled, threshold, and conditional delegation mechanisms can be dynamically and flexibly applied for distributed agent trust management in the mulit-agent systems. Then, a delegation logic framework has to be set up in order to accurately inference each agent request authority based on the collected certificates.

In [2], agent delegation mechanisms can be shown as delegation network. We show one of simple delegation network in Figure 3. In this delegation network, the "nodes → nodes" and "arcs" are corresponding to issuer/subject agent's public keys and associated delegation certificates. Any subject agents are allowed to use their collected certificates $A_a - Certs$ to request services(or resources) from service provider agent.

5.1 Authority Verification Process

The verification process ensure that each issuer agent in the delegation network should have the right authority declaration for re-delegation and all of the $A_a - Certs$ are within the validation period without any revocation status. The certificate revocation issues are very difficult to handle due to the information negation features. Usually a non-monotonic delegation logic was proposed to handle the certificate revocation problem [23]. In this study, we did not explicitly solve the certificate revocation problem.

5.1.1 Verification by Original Authority Issuer

In pure SPKI/SDSI certificate verification, the original authority issuer agent is also the final authority verifier. This issuer agent is usual a service provider that offers the services based on incoming collected authorization certificates. The authority delegation policies are always specified in the original authority issuer agent's rules base to verify the authority validity.

In [23], the authority delegation rule can be shown as follows:

```
H    if    F
```

where H is a head statement and F is a body formula. A clause with empty body is called a *fact*. A *query* takes the form: "F?" where F is a body formula. The issuer of the head statement is also said to be the issuer of the rule. Rule set in the rules base is a finite set of clauses.

Assume that the global X.509 CA is E-Trust and the local X.509 CA for each human h's agent system is h-Trust. So the rule set in Internet bank $Morgan$ rules base are shown as the followings:

Rules

- Morgan delegates the operation *certify* of $I_h - Cert(?ID_h, ?PU_h, Options)$ to E-Trust.
- Morgan delegates the operation *certify* of I_a-Cert($?ID_h\#ID_a$, $?PU_a$, Options) to h-Trust.
- Morgan delegates the operations of *(check,transfer, withdraw, deposit)* to Owner($?ID_h$) on Account(?Acc) if IsAccountOwner($?ID_h$, ?Acc)
- Morgan says PublicKey($?PU_a$) speaks for Name($?ID_h$) if IsPublicKey($?PU_a$, $?ID_h$)
- Morgan says threshold(2,?S, E-Trust says belongTo(?S, BobOrg)) speak for Owner($?ID_h$) on Account(?Acc) if Owner($?ID_h$) delegates the operations *(check,transfer,withdraw,deposit)* to threshold(2,?S, E-Trust says belongTo(?S, BobOrg))
- Morgan says the operations *(transfer,withdraw)* on Account(?Acc) must less than Balance(?Acc)

FACTS

- E-Trust says $I_{Bob} - Cert = (ID_{Bob}, 123456, Options)$.
- Bob-Trust says $I_G - Cert = (ID_{Bob}\#ID_G, 783452, Options)$.
- Bob delegates the *(check, transfer, withdraw, deposit)* operations to agent G on Account(ABC) from 2001/04/05 to 2001/04/07.
- Agent G delegates the operations *(check,withdraw)* to agent threshold(2,(M_1,M_2,M_3)) on Account(ABC) from 2001/04/05 to 2001/04/07.
- Agent M_1 threshold-initiate the operations *(check, withdraw)* to agent M_2 on Account(ABC) from 2001/04/05 to 2001/04/07.
- Agent M_2 threshold-delegates the operations *(check, withdraw)* to agent M_1 on Account(ABC) from 2001/04/05 to 2001/04/07.

Query

Do you allow agent M_1 withdraw(NT$10000) on Account(ABC) on 2001/04/06?

The delegation **rules** in the rule base are configured by the policy maker in the Internet bank *Morgan*. The **facts** are dynamically stored from global and local CAs, authority subject agents, etc. If the authority subject agent did not store the delegation certificate for each re-delegation, then all of the re-delegation certificates have to present to the verifier agent when query is initiated for inference information completeness.

The logic inference in backward-chaining might be executed to answer the above agent M_1 query based on the generic rules and dynamically cumulative facts. These facts are the identity and authorization certificates stored in the verifier agent rule base by associated issuer(or subject) agents.

5.1.2 Verification by TTP

The verification process does not necessarily to be done by the service provider agent. It might be referred to the Trusted Third Part(TTP) for yes/no answer. In this way, all of the rules, facts, and queries are processed in the TTP. The TTP will be responsible for all of the services access control and this verification pattern will reduce the services provider workload. But the TTP must accept the liability challenge due to the inference errors. Of course, the successful evaluation of TTP trustworthiness is also a sufficient condition for the adoption of the TTP verification.

6. AGENT DELEGATION IN FIPA ACL

From the obsolete FIPA agent security management specification [8], we can see that the basic X.509-based identity certificate PKI for agent have not been specified yet, so don't even mention about the agent-oriented PKI for both agent identification and authorization. Thus, there are still lots of works needed to be done before deploying the FIPA-compliant agent system in the real-world applications.

6.1 Our Approach

We are implementing the agent-oriented PKI in the FIPA-OS(Open Source) toolkits [2]. Several new speech-act based communicative acts were proposed to manage both the human/agent identity certificate and human/agent authorization certificate.

6.1.1 Speech-Acts for Identity Certificate

The speech-act performatives for human/agent identity certificate management are certificate registration, query, and revocation:

- $register-I_h-Cert/register-I_a-Cert$: human/agent identity certificate $I_h - Cert/I_a - Cert$ is generated at human/agent local computer and registered at global / local X.509 CA. The $register - I_h - Cert/register - I_a - Cert$ performative is initiated when human/agent

[2] see http://fipa-os.sourceforge.net

is registering the $I_h - Cert/I_a - Cert$ to the agent located at global/local X.509 CA.

- $query - I_h - Cert/query - I_a - Cert$: use human/agent unique symbolic ID to query the human/agent unique public key PU_h/PU_a in global/local X.509 CA for further human/agent authorization certificate $A_h - Cert/A_a - Cert$ generation.
- $store - I_h - Cert/store - I_a - Cert$: global/local X.509 CA uses this performative to store the valid identity certificate in the verifier agent' rule base.
- $revoke - I_h - Cert/revoke - I_a - Cert$: use these performatives to revoke the human/agent identity certificate before the end of validation period.

6.1.2 Speech-Acts for Authorization Certificate

The speech-act performatives for human/agent authorization certificate management are simple certificate delegation, threshold certificate delegation, certificate storing in the verifier rule base, etc.

- $delegate - A_h - Cert/delegate - A_a - Cert$: once the $delegate - *$ performative was initiated, the authority issuer agent must sign the certificate first, then use associated $delegate - *$ performative to notify the authority subject agent for possible further re-delegation.
- $threshold - initiate$: as the authority subject agent receives an authorization certificate $A_a - Cert$ with threshold as conditions, then this agent uses $threshold - initiate$ performative to initiate the authority coordination process with the other subject agents in the threshold subject agent pool.
- $threshold - delegate$: once the agent receives $threshold - initiate$ performative , it might use $threshold - delegate$ to give its consent to the threshold initiation agent.
- $store - A_h - Cert/store - A_a - Cert$: any authority subject agent might use this performative to store the valid authorization certificate in the verifier agent rule base for future quick verification inference.

The reasons that we did not use existing FIPA 20 standard communicative acts(or performatives) for our agent to initiate the certificate management are the simplicity of the communication. Otherwise, we have to recursively embed all of the operators in the agent content field. Besides, the semantics of ACL in this recursive operators are very hard to define. The responsive performative to all of the above initiative operations will be $inform(done)$ in the FIPA ACL standards.

Above suggested performatives for identity/authorization certificate management are pretty subjective. We are still working on the syntax and semantic of these ACL. As we reflect on the difficulty in verifying the semantic issues in ACL [30], the standardization of agent management security still needs further study.

6.1.3 ACL Encoded in XML

The emerging technology to encode the ACL is via XML [14]. The proposed new performatives for agent certificate management in this study are quite easy to embed to the XML DTD file in fipa.acl.rep.xml.std [9]. The ACL encoded in XML only provides the syntax interoperability. So the semantic interoperability feature of ACL in the agent state and in the agent content language must be done in XML/RDF.

6.1.4 Content Encoded in XML/RDF

In FIPA 2000 experiment standard, the agent content language can be encoded in XML/RDF [10]. So the facts and rules stored in our verifier agent's rule base can be expressed in fipa-rdf0 and fipa-rdf1. XML/RDF provides semantic interoperability that gives the agents interpret an RDF data model in the same way.

For example, the fact E-Trust says $I_{Bob} - Cert = (ID_{Bob}, 123456, Options)$ might be shown as:

```
(store-I(Bob)-Cert
   :sender E-Trust
   :receiver Morgan
   :content(
    <?xml version="1.0">
    <rdf:RDF xmlns:rdf="http://www.w3.org/....."
             xmlns:s="http://desp/schema/">

       <rdf:Description ID="Bob Public Key">
         <s:pub-key>123456</s:pub-key>
       <!-- other optional profiles for Bob -->
       </rdf:Description>
    </rdf:RDF>)
    :language fipa-rdf0
    :signature efa23bcd)
```

Signature $efa23bcd$ parameter shows the above entire message was signed by E-Trust private key to enfore the says performative. Recently, a logical interpretation of XML/RDF with inference engine mechanisms were studied extensively.[3] The complete representation and execution of verifier agent's rule base in fipa-rdf0 and fipa-rdf1 for our agent certificate management still need further study.

6.1.5 Further Studies

The agent trust and delegation problem is one of the very promising research areas for multi-agent system infrastructure. If we can not handle the above issues in technology and legal complete manner, then the dream of agent system to serve the entire human society can not be in reality. In our on-going research project, we are still exploring a generic global agent-oriented identity PKI with associated authority certificate, which can support the agent trust and delegation process as well as relevant security, safety, and privacy issues. In the near future, we are expecting the general trust issues for human and agent will be clarified during certificates delegation and verification process to meet those requirements.

7. CONCLUSIONS

We do believe that the agent trust and delegation problem is one of the most important research areas in agent-mediated cyberspace. At this moment, we did not handle all of the human vs. agent trust issues before the agent's authority delegation. Instead, an agent-oriented PKI was proposed to provide identification and authorization trust

[3] http://nestroy.wi-inf.uni-essen.de/xwmf/

management. In this agent-oriented PKI framework, we have identity and authorization certificates operations under different delegation mechanisms, such as chain-ruled, threshold, and conditional, etc. Furthermore, the agent trust and delegation logic was demonstrated in one specific Internet bank example. Finally, we propose some communicative acts for the identity and authorization certificate management and the related XML and XML/RDF encoding concepts were also briefly demonstrated.

8. ACKNOWLEDGMENTS

This work was supported in part by Taiwan NSC(National Science Council) under grants No. NSC 89-2213-E-004-002. Some of the ideas in this paper come from the constant discussion with the students in ENT(Emerging Network Technology) lab. Besides, figures 1, 2, 3 in this paper are also drawn by the students in the ENT lab.

9. REFERENCES

[1] Abadi, M., Burrows, M., and Lampson, B., A Calculus for Access Control in Distributed Systems. *ACM Transactions on Programming Languages and Systems,* 15, 4, (Sep. 1993), 706-734.

[2] Aura, Tuomas, On the Structure of Delegation Network. *Proc. 11th IEEE Computer Security Foundations Workshop,* IEEE Computer Society Press, June 1998, 14-26.

[3] Aura, Tuomas, Distributed Access-Rights Management with Delegation Certificates. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects* LNCS 1603, Springer-Verlag, 1999, 213-238.

[4] Castelfranchi, Cristiano and Falcone, Rino, Trust and Control: A Dialectic Link, *Applied Artificial Intelligence,* 14:799-823, 2000.

[5] Clarke, Dwaine, et al., Certificate Chain Discovery in SPKI/SDSI(draft), MIT Lab for Computer Science, Nov. 1999.

[6] Ellison, M. Carl, et al., SPKI Certificate Theory, RFC 2693, Internet Society, 1999,

[7] *FIPA 97 Specification Part 2, Version 2.0, Agent Communication Language,* 1998.

[8] *FIPA 98 Specification Part 10, Version 1.0, Agent Security Management,* 1998.

[9] *FIPA ACL Message Representation in XML Specification,* 2000.

[10] *FIPA RDF Content Language Specification,* 2000.

[11] Ford, Warwick and Baum, S. Michael, *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures & Encryption,* Prentice Hall, 1997.

[12] Gerck, Ed, *Toward Real-World Models of Trust: Reliance on Received Information,* http://www.mcg.org.br/trustdef.htm.

[13] Gladney, H., Safe Deals Between Strangers, *IBM Research Technical Report(draft),* IBM Almaden Research Center, August, 1999.

[14] Grosof, N. Benjamin and Labrou, Yannis, An Approach to using XML and a Rule-based Content Language with an Agent Communication Language, *IBM Research Report,* RC 21491(96965) 28 May 1999.

[15] He, Qi, Sycara, Katia, and Finin, Tim, Personal Security Agent: KQML-Based PKI. *Proceedings of the Second International Conference on Autonomous Agents,* May 1998.

[16] Heckman, C. and Wobbrock, O. J., Liability for Autonomous Agent Design, *Autonomous Agents 98,* Minneapolis, MN USA, 1998, 392-399.

[17] Herzberg, A., et al., Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers, *2000 IEEE Symposium on Security and Privacy,* 2000, 2-14.

[18] Jennings, R. N., Sycara, K., and Wooldridge M., A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems,* 1, 7-38, 1998.

[19] Josang, A., Pedersen, G. Ingar, and Povey, D., PKI Seeks a Trusting Relationship, *Proceedings of ACISP 2000,* Brisbane, Australia, July 2000.

[20] Kimbrough, O. S. and Moore, A. S., On Automated Message Processing in Electronic Commerce and Work Support Systems: Speech Act Theory and Expressive Felicity, *ACM Tran. on Information Systems,* Vol. 15, No. 4, October 1997, 321-367.

[21] Lampson, B., Abadi, M., Burrows, M., and Wobber, E. Authentication in Distributed Systems: Theory and Practice. *ACM Trans. Computer Systems,* 10, 4 (Nov. 1992), 265-310.

[22] Li, N., Grosof, N. B., and Feigenbaum. J., A Logic-based Knowledge Representation for Authorization with Delegation, *IBM Research Report,* RC 21492(96966) 28 May 1999.

[23] Li, Ninghui, Grosof, Benjamin, and Feigenbaum, Joan, A Practically Implementable and Tractable Delegation Logic, *IEEE 2000 Symposium on Security and Privacy,* 2000.

[24] Moukas, A., Guttman, R., Zacharia, G., and Maes, P., Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective. *International Journal of Electronic Commerce, Vol. 4 , No. 3*, Spring 2000.

[25] Nwana, S.H., et al., Agent-Mediated Electronic Commerce:Issues, Challenges and some Viewpoints. *Proceedings of the Second International Conference on Autonomous Agents 98*, 1998, 189-196.

[26] Rivest, Ron and Lampson, Bulter, SDSI- A Simple Distributed Security Architecture, http://theory.lcs.mit.edu/ cis/sdsi.html.

[27] Tschudin, F. C., Mobile Agent Security. *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet,* Springer-Verlag, 1999, 431-445.

[28] Wen, Wu and Mizoguchi, Fumio, An Authorization-Based Trust Model for Multiagent Systems, *Applied Artificial Intelligence,* 14:909-925, 2000.

[29] 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).* May 1999, Seattle, Washington, pp. 149-161.

[30] Wooldridge, M., Semantic Issues in the Verification of Agent Communication Languages, *Autonomous Agents and Multi-Agent Systems,* 3, 9-31(2000).