DOI:10.3745/JIPS.2007.3.2.082

A Delegation Model based on Agent in Distributed Systems

Kyu Il Kim*, Joo Chang Lee*, Won Gil Choi*, Eun Ju Lee*, and Ung Mo Kim*

Abstract: Web services are the new building block of today's Internet, and provides interoperability among heterogeneous distributed systems. Recently in web services environment, security has become one of the most critical issues. The hackers attack one of fragile point and can misuse legitimate user privilege because all of the connected devices provide services for the user control and monitoring in real time. Also, the users of web services must temporarily delegate some or all of their rights to agents in order to perform actions on their behalf. This fact risks the exposure of user privacy information. In this paper, we propose secure delegation model based on SAML that provides confidentiality and integrity about the user information in distributed systems. In order to support privacy protection, service confidentiality, and assertion integrity, encryption and a digital signature mechanism is deployed. We build web service management server based on XACML, in order to manage services and policies of web service providers.

Keywords: *XACML, SAML, Agent*

1. Introduction

Web services are the new building block to today's Internet, and provides interoperability among heterogeneous distributed systems. But web service requires still user's access availability for access to service resources and security reliability. User availability must be provided the services without complicate process through only once authentication and user information doesn't exposure privacy information without consent.

Also, the users of web services must temporarily delegate some or all of their rights to agents in order to perform actions on their behalf. Previous researches proposed the multiplicity of authentication technologies like Kerboros[2], Password and PKI structure to solve this problems. But this mechanisms need to be additional building to the client because they exchanges non-XML message with user authentication.

In distributed environments, this paper proposes delegation model based on SAML in case the user use secure web service.

SAML is an XML standard for exchanging authentication and authorization data between security domains. SAML is a product of the OASIS Security Services Technical Committee. One of the major goals for SAML is Single Sign-On(SSO)[8], the ability of a users to authenticate in one domain and use resources in other domains without re-authenticating. SAML has become the definitive standard underlying many web SSO solutions in the enterprise identity management problem space. SAML enable web SSO through the exchange of an authentication assertion from first site to second site.

But SAML don't provide the privilege delegation to the other users. We propose delegation model based on extended SAML. Also, we describe the detail approach mechanism by Access Control Server (ACS).

2. Backgrounds

2.1 SAML

SAML[3] defined the syntax and processing semantics of assertions made about a subject by a system entity. SAML assertions and protocol messages are encoded in XML and use XML namespaces. They are typically embedded in other structures for transport, such as HTTP POST request or XML-encoded SOAP (Simple Object Access Protocol) messages. SOAP is a protocol for exchanging XML-based message over computer networks, normally using HTTP. SAML specification defines three different kinds of assertion statements that can be created by a SAML authority. All SAML-defined statements are associated with a subject. SAML specifications define the below components:

- *Core* (the syntax and semantics for XML encoding assertion and request/response protocols)
- *Bindings* (protocol binding for the use of request / response messages)
- *Profiles* (for embedding and extracting SAML

Manuscript received December 7, 2007; accepted December 27, 2007.
This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-C1090-0801-0028) and this work was supported in part by the Ubiquitous Autonomic Computing and Network Project of 21st Century Frontier R&D Program

Corresponding Author: Kyu Il Kim
***** Department of Computer Engineering, SungKyunKwan University
Engineeringt Building 27309 300 ChunChun Dong, JangAn Gu, Suwon, KyungGi Do, Korea
{kisado, lordeath, wonkiler, eunjoo, umkim}@ece.skku.ac.kr

assertions in a framework or protocol)

- *Authentication context*(the syntax for the definition of authentication context initial list)
- *Conformance*
- *Metadata*
- *Security Consideration*

In order to achieve web single sign-on, SAML must use request/response protocols based on assertion. SAML defines three different types of assertions:

- *Authentication:* The assertion subject was authenticated by a particular means at a particular time.
- *Attribute:* The assertion subject is associated with the supplied attributes.
- *Authorization Decision:* A request to allow the assertion subject to access the specified resource has been granted or denied.

SAML lacks delegation capabilities. However, SAML provide inherent extensibility to create our delegation assertion.

2.2 XACML

XACML (eXtensible Access Control Markup Language)[1][6] is the result of a recent OASIS standardization effort proposing an XML-based language to express and interchange access control policies. XACML was designed to accommodate most system needs. At its core, XACML defines the syntax for policy language and the semantics for processing those policies. There is also a request and response format to query the policy system and semantics for determining applicability of policies to requests. The request and response formats represent a standard interface between a Policy Enforcement Point (PEP) that issues requests and handle responses and a Policy Decision Point(PDP) presents standard behavior when processing policy. Figure1 illustrates XACML overview. This is based on policy framework definitions used in the IETF[7].

- *Combination policy support*. XACML provides a method for combining policies independently specified. Different entities can then define their policies on the same resource. When an access request on that resource is submitted, the system has to take into consideration all these policies. XACML defines three elements for the specification of access control policies: Rule, Policy, and PolicySet. The Rule element corresponds to the traditional concept of authorization: it defined who can access to which resource and under which conditions. The Policy element consists of a set of rules and specifies how to combine the results of their evaluation. Finally, the PolicySet element contains a set of Policy or PolicySet.

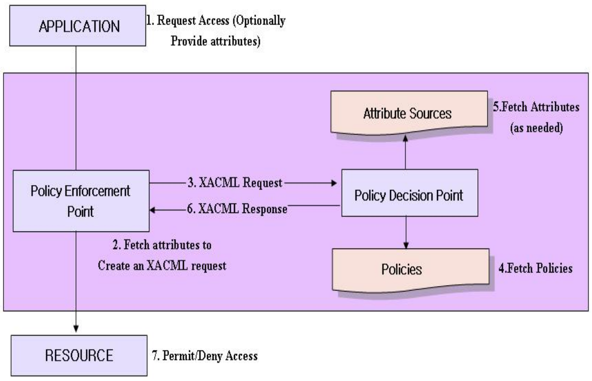


Fig. 1. XACML Overview

- *Combining algorithms support*. Since both a Policy and a PolicySet element can contain multiple policies or rules, each of which can evaluate to different access control decisions, XACML needs to define a method for reconciling such decisions. XACML supports different combining algorithms, each representing a way of combining multiple decisions into a single decision. To this purpose, XACML defines two attributes, namely RuleCombiningAlgId, and PolicyCombiningAlgId. The first attribute indicates a method for combining the individual results obtained from the evaluation of a set of rules. The second attribute indicates a method for combining the individual results of evaluation of a set of policies.
- *Attribute support*. XACML supports the definition of policies based on properties (attributes) associated with subjects and resources other than their identities. This allows the definition of powerful policies based on generic properties associated with subjects (e.g. name, address and occupation) and resource. To this purpose, XACML provides two elements, namely SubjectAttriubuteDesignator and ResourceAttribute Designator that together with SubjectMatch and ResourceMatch elements allow identifying a particular subject and resource attribute, respectively.
- *Operators support*. XACML includes some built-in operators for comparing attribute values and provides a method of adding non-standard functions.
- *Multiple subject*. XACML allows the definition of more than one subject relevant to a decision request.
- *Policy distribution support*. Policies can be defined by different parties and enforced at different enforcement points. Also, XACML allows one policy to contain or refer to another.
- *Implementation independency*. XACML provides and abstraction-layer that isolates the policy writer from the implementation details. This means that different implementations should operate in a consistent way, regardless of the implementation itself. As we will see later on, XACML defines a canonical form for the request and response, called XACML context.

- *Obligations support*. XACML provides a method for specifying actions, called obligations that must be fulfilled in conjunction with the policy enforcement. The element that provides this feature is the Obligation element.

3. Related Works

Navarro, et al[5] describe SAML assertion for constrained delegation. In order to support delegation, they extend SubjectStatement to SubjectDelegationStatment which is not supported by the SAML1.1/2.0 Assertion specifications. J. Wang and D. Del Vecchio [4] also extend SAML to support delegation. They propose verification rules for delegation assertions relying on a WS-Security X.509 Signature[9]. However, they must verify an ordered delegation chain to prove a trust relationship between a delegator and a web service portal when the indirect delegation is applied. Y. J. Hu[10] presents a method for creating agent systems, which is based on Public Key Infrastructure (PKI). He considers various kind of delegation such as chain-ruled delegation, threshold delegation and conditional delegation. However, this approach does not consider mobile agent system or heterogeneous multi-agent systems. Navarro, et al [14] present an access control framework for a mobile agent platform, which is based on RBAC. In this framework, the Authorization Manager (AM) is exploited, to manage the delegation of authorization, and issuing of authorization certificates. However, they only consider agents complying with its local policy.

We exploit the authentication/delegation authorities to omit the step of verifying and ordered delegation to verify a trust relationship. An important contribution of our work is applicable to delegation for web service using all kinds of agents.

4. Delegation Model based on SAML

In figure2, we describes secure delegation model based on SAML in the distributed systems. By the internet network, User Agent processes the works instead of the users. UA requests the role to ACL for web service. At this time UA request the role as a SAML *AuthorizationDecision* request message. ACL this message is received by the UA is consist of structure based on the XACML. XACML can represent the functionalities of most policy representation mechanisms and has stand extension points for defining new function, data types, policy combining logic. ACL carry out the following procedure in order that ACL grants the proper role to UA by the current user's situation.

Policy Administration Point (PAP) writes polices and policy sets and make them available to the Policy Decision Point (PDP). PAP module retrieves the policies applicable

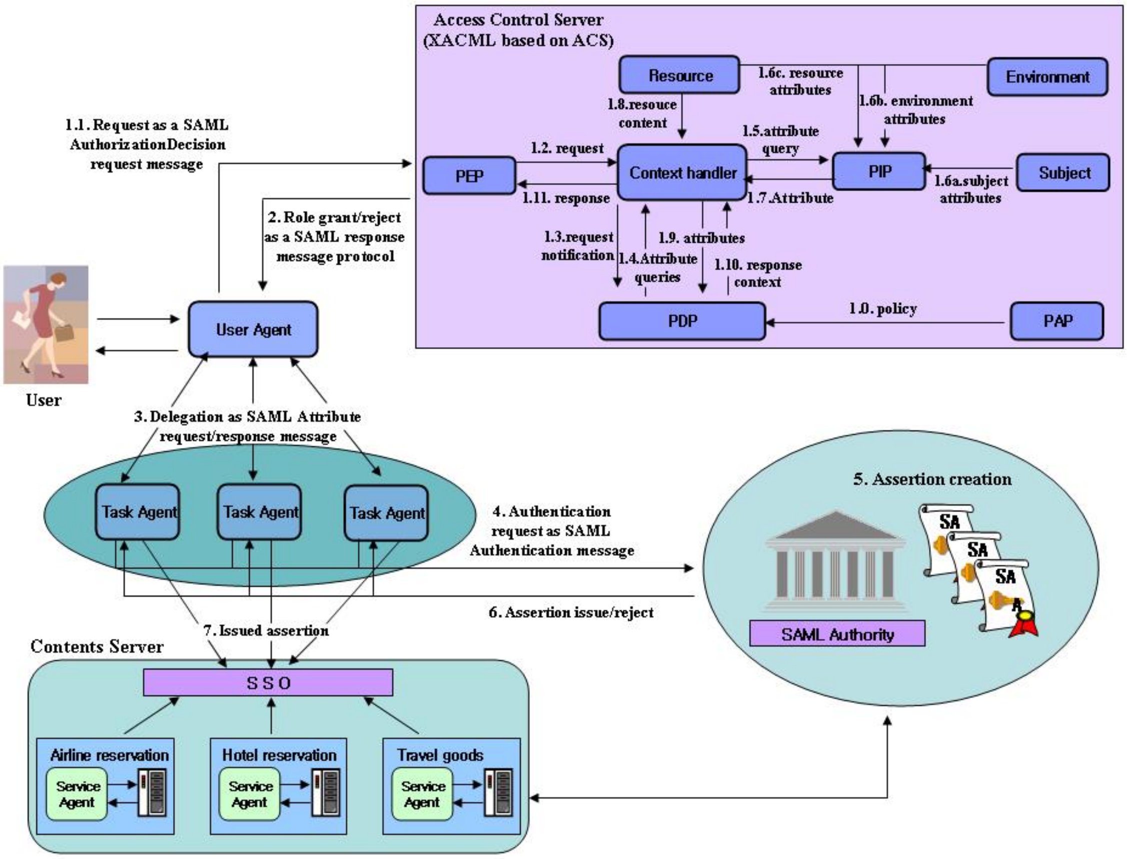


Fig. 2. Delegation Model based on SAML

to a given access request and returns them to the PDP module. And PDP module receives an access request and interacts with the PAP that encapsulates the information needed to identify the applicable policies. User Agent sends a request for access to the Policy Enforcement Point (PEP). PEP is a module that enforces the access decision taken by the decision point.

The PEP sends the request for access to the context handler in its native request format, optionally including attributes of the subjects, resource, action and environment. The context handler constructs an XACML request context and sends it to the PDP. The PDP requests any additional subject, resource, action and environment attributes from the context handler.

The context handler requests the attributes from a Policy Information Point (PIP). PIP module provides attribute values about the subject, resource, and action. The PIP obtains the requested attributes. The PIP returns the requested the requested attributes to the context handler. Optionally, the context handler includes the resource in the context. The context handler sends the requested attributes and the resource to the PDP. The PDP returns the response context to the context handler. The context handler translates the response context to the native response format of the PEP. The context handler returns the response to the PEP. The PEP fulfills the obligations. If access is

Permitted, then the PEP sends the role permitted as a SAML response message protocol to User Agent, Otherwise, it denies access.

SAML *AuthzDecision* statement is as follows in figure3.ID is an identifier for the request. *ID* is the values of the ID attribute in a request and the InRespnoseTo attribute in the corresponding response must match. *Version* is the version of this request and it is the identifier for the version of SAML defined in this specification is SAML V2.0. *IssueInstant* is the time instant of issue of the request. *Subject* name is requesting the principal name to ACL. *Resource* is the service factor for which authorization is requested. *Action* is the privilege for which authorization is requested. *Signature* is user agent that authenticates the requester and provider message integrity. All SAML protocol defines request and response messages that signed using XML signature [13]. Figure 4 shows the example of the *AuthzDecision* request message.

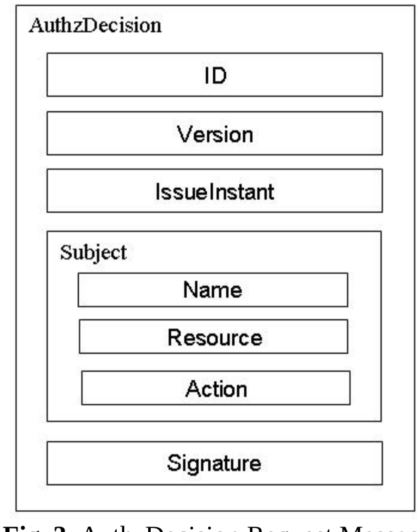
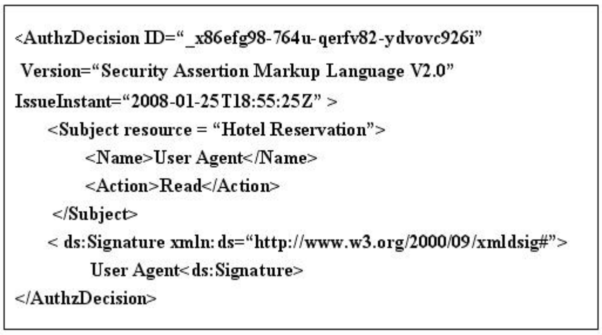


Fig. 3. AuthzDecision Request Message

ACL verifies *AuthzDecision* message and grant/reject the role as a SAML response message to UA.
Within response message, *status attribute* is a result representing the status of the corresponding request. According to the role hierarchy [11], UA Assigned by

The ACL delegates the equal role or the lower role to Take Agent (TA) as SAML attribute request/response message. *Attributestatement* request/response message include delegation element. For more delegation will explain detail in the SAML assertion structure.



```
<AuthzDecision ID="_x86efg98-764u-qerfv82-ydvovc926i"
 Version="Security Assertion Markup Language V2.0"
IssueInstant="2008-01-25T18:55:25Z" >
    <Subject resource = "Hotel Reservation">
        <Name>User Agent</Name>
        <Action>Read</Action>
     </Subject>
    < ds:Signature xmln:ds="http://www.w3.org/2000/09/xmldsig#">
         User Agent<ds:Signature>
</AuthzDecision>
```

Fig. 4. AuthzDecision SAML Message Exam

Task Agent works the task instead of the UA. Task Agent confirms User Agent's role and delegation. And TA requests the authentication to SAML authority to receive authentication. SAML authority analyzes the status by the authentication message and issues the assertion or reject.

In figure5, SAML assertion structure is follows.

ID, *Version* and *IssueInstant* equal *AuthzDecision* message structure too. *Issuer* is the SAML authority that is making the claim in the assertion. *Condition* indicates the valid duration of delegation assertion using *NotBefore* and *NotOnOrAfter* attributes. In the *AttributeStatment*, *Subject* Name is Task agent name. Delegation indicates whether the agent that receives this delegation assertion, is allowed to delegate the right to another agent. For example, if this is "true", delegation is granted. *Consent* indicates whether the SAML authority obtains the User Agent consent.

In *AuthenticationStatement*, Subject name is Task Agent and *AuthIstant* is the time which the authentication took place. *AuthnContext* is the context used by the authenticating up to and including the authentication event that yielded this statement. *AuthnContext* contains an authentication context class reference. *EncryptedData* is assertion data for cryptographic as defined by the XML encryption syntax [12] and *EncryptedKey* is decryption keys wrapped, as defined by the XML encryption syntax too.

SAML authority confirms the assertion and issues the assertion with digital signature as SAML authority private key. If digital signature corresponds to task agent pubic key, these assertion certificates digital signatures as SAML authority private key. TA access to SSO web service by the assertion.

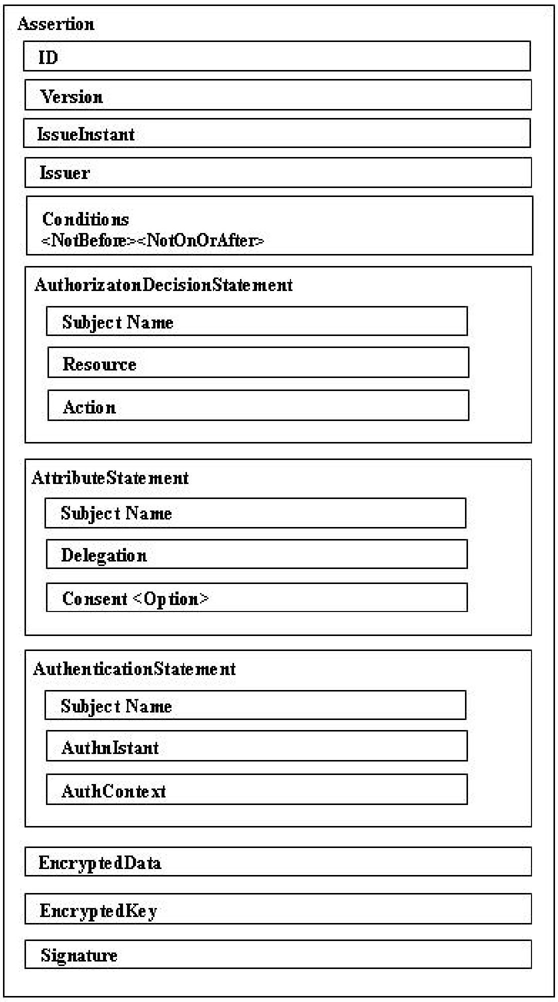
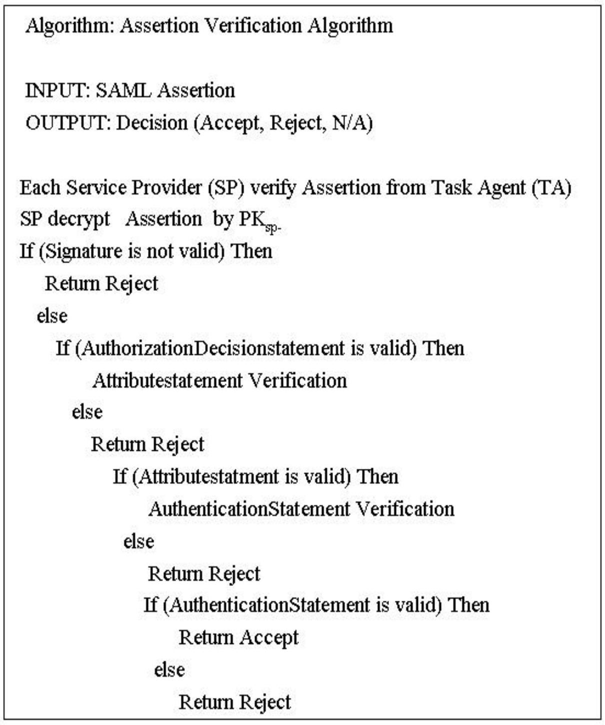


Fig. 5. Secure Delegation SAML Assertion

5. Implementation

We request (response) the SAML message before issuing the assertion by SAML authority. Hence we reduced the assertion creation time than the existing algorithms. And yet, SAML message can certificate message protocol because is included the digital signature and trust information. Also UA decreases the assertion verification time too because delegate the privilege to Task Agent before issuing the assertion. This means that Task Agent is confirmed only once authentication to SAML authority.

For example, if Task Agent delegates the privilege from TA_1, TA_2, TA_3… to TA_i, existing mechanism is received the authentication not only TA_1, TA_2 but also T_i to all SAML authority. Even though we again delegation to Task Agents, we decreases the creation time because is checked only once authentication by SAML authority. Figure 6 is the assertion verification algorithm. TA encrypts the assertion as Service Agent's public key and send. SA decrypts the assertion as SA's private key and confirms digital signature. If digital signature is not valid, SA sends the reject message to TA. And SA checks whether SA verifies the three statements valid or not. If the three statements permits access to TA or denies access to TA.



```
Algorithm: Assertion Verification Algorithm

INPUT: SAML Assertion
OUTPUT: Decision (Accept, Reject, N/A)

Each Service Provider (SP) verify Assertion from Task Agent (TA)
SP decrypt  Assertion  by $PK_{sp}$.
If (Signature is not valid) Then
   Return Reject
  else
    If (AuthorizationDecisionstatement is valid) Then
        Attributestatement Verification
      else
        Return Reject
          If (Attributestatment is valid) Then
              AuthenticationStatement Verification
            else
              Return Reject
              If (AuthenticationStatement is valid) Then
                  Return Accept
                else
                  Return Reject
```

Fig. 6. Assertion Verification Algorithm

6. Security Analysis

We analyze a proposed framework with respect to security analysis as following:

(1) *Confidentiality*: confidentiality is one of the cornerstones of information security for ensuring information that is accessible only to requests authorized to have access. In the proposed framework, DA issues delegation assertion in order to support secure and efficient delegation of rights among agents. We ensure confidentiality by means of encrypted tags defined in delegation assertion (see the Figure 4). Only dedicated recipient is able to decrypt *EncryptedData* because *EncryptedKey* is encrypted by public key of recipient.

(2) *Integrity*: integrity means that assets can be modified only by authorized parties or only in authorized ways. In the proposed framework, delegation assertion is only manipulated by DA, and contains digital signature value as the content of Signature element. Signature element also contains information such as digest value and method in order to support the integrity of delegation assertion. We ensure integrity of delegation assertion using these digest information.

(3) *Replay attack*: A replay attack is a form of network attack in which a valid data transmission is maliciously repeated or delayed by an adversary who intercepts the data. An attacker wants to impersonate the legal request to web service provider using a stolen delegation assertion from legal agent. However, it is impossible because delegation assertion contains digest information in order to prevent fabrication, and is valid during the period defined in *NotBefore* and *NotOnOrAfter* attributes of the Condition element.

(4) User's privacy: In the pervasive computing environments, to protect user's privacy is very important because an attacker may expose information related user's privacy to the outside without authentication. We ensure user's privacy by means of public key cryptography method. In the proposed framework, AA only knows who P is because subject element of delegation assertion is encrypted by AA's public key. Only dedicated web service provider is able to decrypt the *EncryptedAttribute* element of delegation assertion because *EncryptedAttribute* element is encrypted by public key of dedicated web service provider.

7. Conclusion

This paper has proposed a delegation framework for ensuring security of web service in distributed systems. However, it requires the system that processes the task work instead of the users. At this time, the user must delegate certainly user's privilege to the agents. Therefore we presented a delegation framework which is managed by user with assistance of authentication/delegation authorities. Also, we expanded SAMLv2.0 based on XACML and provided confidence mechanism by the delegation. If User Agent delegates the privilege to TA_1, TA_2, and TA_i, current model verified all agent assertion. And in case of the assertion creation all agents are confirmed authentication and delegation to authentication authority. But proposed model improved efficiently the assertion creation and verification time because user agent delegate to TAs before assertion issuing to SAML authority.

Future work will be applied to mobile devices with limited memory capacity.

References

[1] OASIS "eXtensible Access Control Markup Language (XACML)V2.0", OASIS Standard, 1 February 2005

[2] B.Clifford Neuman and Theodore Ts'o, Kerberos, An Authentication Service for Computer Networks, IEEE Communications, September 1994 pp33-38

[3] OASIS "Profile for the OASIS Security Assertion Language (SAML)V2.0" OASIS Standard, 15 March 2005

[4] Jung Wang, David Del Vecchio, Marty Humphery, Extending the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services, In Proceedings of the IEEE International Conference on Web Services, 2005

[5] G.Navarro, B.S.Firozabadi, E.Rissanen and J.Borrell, Constrained delegation in XML-based Access Control and Digital Rights Management Standards, Communication, Network, and Information Security 2003.

[6] C.A Ardagan, E.Damiani, S.De Capitani di Vimercati, P.Samarati, XML-based Access Control Language, 2004

[7] R. Yavatkar, D. Pendarakis, and R. Guerin, A Framework for Policy-based Admission Control, IETF Informational Standard, RFC 2753, January 2000.

[8] B.Pfitzmann, B.Waidner, Token-based web Single Sign-On with Enabled Clients, IBM Research Report RZ 3458(93844), Nobember 2002

[9] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder and F. Siebenlist, X.509 Proxy Certificates for Dynamic Delegation, 2004

[10] Y. J Hu, Some thoughts on agent trust and delegation, In Proceeding of the fifth International Conference on Autonomous Agents, 2001.

[11] R.Sandhu, E. Coyne, H. Feinstein, and C.Youman, Role-Based Access Control Models, IEEE Computer, February 1996

[12] XML Encryption Syntax and Proceeding http://www.w3.org/TR/2002/REC-xmlenc-core-20021210

[13] XML Signature, http://www.w3.org/TR/xmldsig-core

[14] G. Navarro, J. A. Ortega-Ruiz, J. Ametller, S. Robles, Distributed Authorization Framework form Mobile Agents, LNCS Mobility Aware Technologies and Applications, 2005

Kyu Il Kim

He received his B.S. and M.S. degrees in Electrical and Computer Engineering from the Wonkwang University and the Sungkyunkwan University in 2003 and 2005, respectively. He is currently pursing his Ph.D degree in Computer Engineering at Sungkyunkwan University. His research interests include database security, access control, XML security, and ubiquitous computing.

Joochang Lee
He received his B.S. degree in Information and Communication Engineering from the Sungkyunkwan University in 2007. He is currently a master student in Computer Engineering at the Sungkyunkwan University. His research interests include privacy preserving data mining, database security, and data privacy.

Won Gil Choi
He received his B.S. degree in Information and Communication Engineering from the Sungkyunkwan University in 2007. He is currently a master student in Computer Engineering at the Sungkyunkwan University. His research interests include data mining, XML mining, and ubiquitous computing.

Eunju Lee
She received her B.S. degree in Electronics Engineering from the Korea Polytechnic University in 2007. She is currently a master student in Computer Engineering at the Sungkyunkwan University. Her research interests include data mining, XML mining, database security, and ubiquitous computing.

Ung Mo Kim
He has been a professor of the School of Information and Communication Engineering at the Sungkyunkwan University since 1994. He received his B.S. in Mathematics from the Sungkyunkwan University, M.S. in Computer Science from the Old Dominion University in 1981 and 1986, respectively, and his Ph.D. degree in Computer Science from Northwestern University in 1990. His research interests include data mining, database security, web databases, information retrieval, and spatial databases.