International Journal of Electronic Commerce Studies
Vol.5, No.1, pp. 39-62, 2014
doi: 10.7903/ijecs.1157

TECHNOLOGY OF FEDERATED IDENTITY AND SECURE LOGGINGS IN CLOUD COMPUTING

Takashi Shitamichi
Tokyo Denki University
5 Senjyu-Asahicyo Adachi-ku, Tokyo, 120-8551 Japan
11udc01@ms.dendai.ac.jp

Ryoichi Sasaki
Tokyo Denki University
5 Senjyu-Asahicyo Adachi-ku, Tokyo, 120-8551 Japan
sasaki@im.dendai.ac.jp

ABSTRACT

Federated services are becoming widely implemented at many sites in multiple domain networks for cloud computing across many industry segments. New technology is required not only for federated authentication, but also for services operating distributed attributes, which are both static and dynamic. In addition to the technology, the sites that provide services across multiple domain networks are required to store every log as audit trails. This paper focuses on SAML and ID-WSF, which are the technology and the architecture for identity management and secure web services, discusses deployments and problems in the real world, then proposes a fast and safe technology that extends the ID-WSF for services and logs. To verify the effectiveness of the proposed technology and architecture, the latencies of SAML SSO that exchange SOAP messages are measured and considered in a cloud computing environment.

Keywords: Federation, Identity, Authentication, SAML, ID-WSF, Cloud Computing, Log

1. INTRODUCTION

Along with the wide deployment of cloud computing in many fields, the number of network services not only completed within the same network domain, but also federated with multiple sites among network domains has been increasing. In the case of federated services, federated authentication

among multiple sites is necessary. Some federated authentication standards are Security Assertion Markup Language (SAML), OpenID, and Open Authorization (OAuth)[1,2,3].

A set of aggregated personal information held by each site providing federated services is called an "identity" and a set of personal information of attributes associated by federated authentication is called a "federated identity".

To use the personal information of attributes of a federated identity, protocols that consider privacy protection and security are necessary. In addition to SAML, a transmission method for attributes using Shibboleth extended by SAML and the Liberty Identity Web Service Framework (ID-WSF), which applies the mechanism of identity verification using SAML assertion, have been specified[4,5].

Some companies that develop services on cloud computing provide an Application Program Interface (API) to perform federated authentication of their users' information systems. One example of enterprise usages of cloud computing is Google Apps, which uses the SAML API known as "SAML Single Sign-On (SSO) Services for Google Apps"[6]. Another example is Salesforce.com, which provides the SAML API known as "Single Sign-On with SAML on Force.com"[7]. As enterprise usages of cloud computing, SAML has gained a favorable reputation for federated authentication and identity.

Although the technology for federated identities such as SAML has been deployed in many places, the technology for usage of personal information of attributes has not been sufficiently studied from the viewpoint of both speed and safety. Personal attributes include not only the information of "static attributes", such as name, address, birth date, gender, etc., but also the information of "dynamic attributes", such as GPS location, blood pressure, and pulse, etc.[8]. In past studies of federated identity technology for managing static attributes, the technology of roaming methods for a user's device has been developed to shorten the elapsed time of authentication[9]. But no work has been done on the federated identity technology that manages dynamic attributes, which are accumulated at every moment of every day, with consideration of speed and safety. If a site that manages the information of attributes cannot quickly transfer that information to another site during a user interaction, the user experience can be severely affected. If the site is located in an area that has low network latency, the problem of a long elapsed time of transferring the attributes may not occur. However, if the site is in a cloud computing data center that is geographically located at a far distance, for example, at the opposite side of

the earth, the latency is expected to be so high that the Round Trip Time (RTT) is long. This condition can also have a negative impact on the user experience.

These scenarios have led to the study of a new architecture that extends over the current specification of SAML and ID-WSF to manage both static attributes and dynamic attributes with speed and safety. In this paper, the real performance of some sites that deploy SAML in the current cloud computing environment is measured and examined as preparation work to verify the effectiveness of the new architecture.

2. FEDERATED AUTHETICATION USING SAML

SAML was developed in the early 2000s as a framework for exchanging security information through a network. Since then, it was enhanced to SAML2.0 to include the specification of the Identity Federation Framework (ID-FF) specified by the Liberty Alliance Project, and consequently SAML2.0 has been widely deployed in many kinds of Internet and intranet services. In the cloud computing environment, numerous actual enterprise information systems use SAML for federated authentication and identity, as stated in Section 1. In this section, the mechanism of the Single Sign-On model using SAML is discussed and considered for deployment in the cloud computing environment.

2.1 Features of SAML Federated SSO and Profiles

One of the features of a SAML federated SSO is its ability to establish the "Circle of Trust" (CoT) and then enable the user to perform an SSO using a "User Agent" (UA) such as a web browser. The architecture of SAML SSO for establishing a CoT is illustrated in Figure 1. In this figure, "SP" is the Service Provider and "IdP" is the Identity Provider.

Each site independently owns its user accounts, and each account of the same user in both sites is associated with an identifier that is called an “opaque handle”. This technology enables each site to independently keep the user identity and protect user privacy because no common information of the identity is shared. In the architecture example shown here, two sites enable SAML SSO. One of them is the IdP and the other is the SP. IdP is the authority that creates claims, which are elements of the token, such as name and age, and operates the Security Token Service (STS). SP uses the claims, specifies the user, and provides an application service. Figure 1 shows the basic concepts of SAML SSO using the UA, the IdP, and the SP. The figure is called a "three corner model".

The SAML specification states Assertions, Protocols, and Bindings. In addition, Profiles define some usage cases, one of which is the SSO. For example, the SSO profile using a web browser defines the following three kinds of bindings:

(a) SP-Initiated SSO: Redirect/POST Bindings

(b) SP-Initiated SSO: POST/Artifact Bindings

(c) IdP-Initiated SSO: POST Bindings.

In these three kinds of profiles, (b) is receiving attention because it considers security using Simple Object Access Protocol (SOAP) messaging. The sequence of (b) in the SAML specifications is illustrated in Figure 2[10].

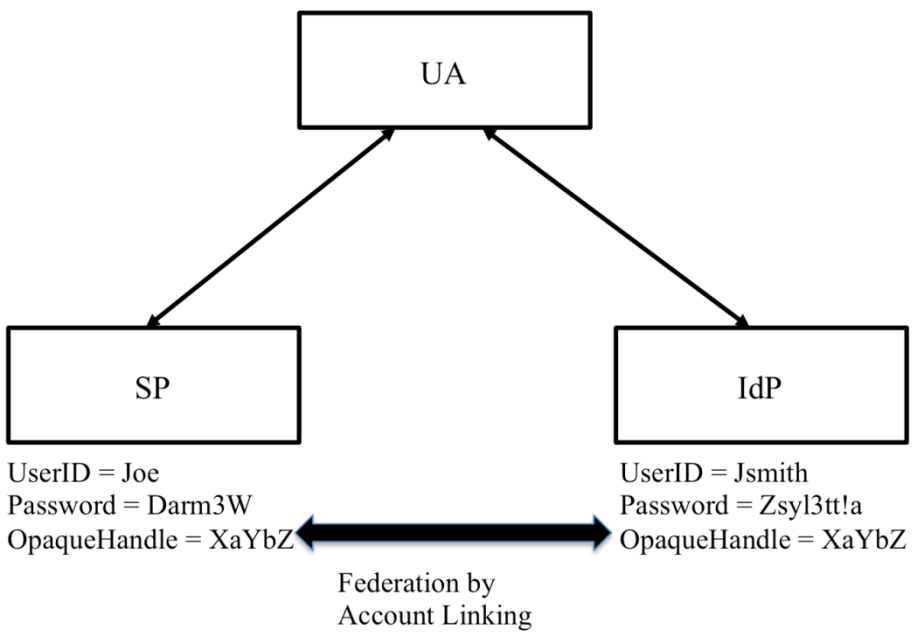

Figure 1. SAML federated SSO architecture

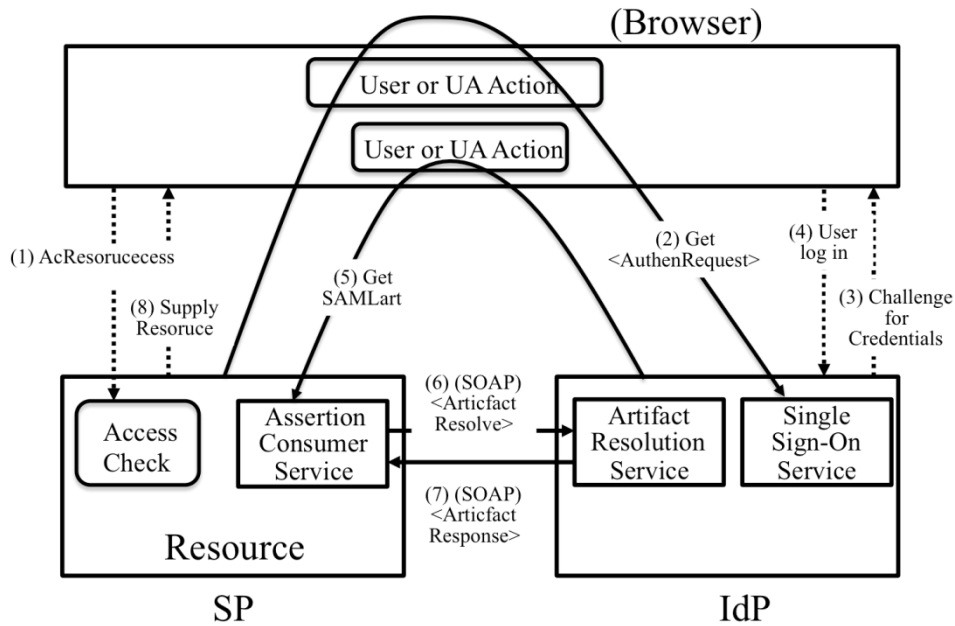

Figure 2. SAML SSO POST/Artifact Bindings

2.2 Consideration of SAML SSO in Cloud Computing Environment

Problems of deployment of enterprise information systems occur in cloud computing environments that are spread worldwide. Much data including personal information are exchanged across the trust boundary managed by an enterprise, so it is supposed that privacy protection is a legal issue[11]. Moreover, it is supposed that the latency of messaging between distant places is a technical issue. For example, the latency of two sites communicating within Tokyo is much lower than the latency between Tokyo and Latin America. This results in a different user experience. According to one investigation, people using services on the web feel annoyance if the web browser takes more than three seconds to display the web page: 47% of users expect the web page to be displayed in less than two seconds, and if it takes more than three seconds, 40% of them leave the website[12]. Therefore, user experience on the web is very important.

As described in Section 2.1, deployment of Artifact Bindings is considerable among the three profiles from the viewpoint of security of SAML SSO service in the cloud computing environment. Latency may be a problem between a user's web browser and service sites. In addition, another possible latency of SOAP messaging is between the IdP and the SP in the case of Artifact Bindings. So, latency negatively affecting user

experience is a real possibility. This means that technology that lowers the latency between two or more sites is necessary to enhance the user experience.

3. PROBLEMS OF TECHNOLOGY TO OPERATE ATTRIBUTES

Many consumer services on the Internet are up and running these days. Some of them, such as Facebook and Twitter, provide unique mechanisms that aggressively manage attributes using APIs. Those consumer services most often use OAuth, which is a specification of authorization to operate attributes. Another specification also standardizes a new specification of OpenID Connect based on OAuth 2.0 for consumer services. However, consideration and experimental proof from the viewpoint of security have not been sufficient and challenges still exist. One of the reasons that service providers have not resolved the security issues is that they prefer launching new services as soon as possible to verify the security mechanism, and so they often use the Representational State Transfer (REST) architecture for many consumer services, although the security mechanism of REST architecture has not been examined adequately[13,14].

A technology to operate attributes based on SOAP is the ID-WSF, which uses the SAML assertion to express security contexts, as specified by the Liberty Alliance Project in 2003 after many security researchers and engineers for web services completed their reviews.

3.1 ID-WSF

ID-WSF is a specification that securely exchanges the information of attributes among multiple sites with the user's consent. To assure safety in a structure, ID-WSF is based on SOAP and specifies the security mechanism. In particular, ID-WSF defines the combinations of confidentiality and the integrity of messages and specifies the classification of security with an identifier called the "security mechanism ID". Of the many combinations, one can do the following to ensure the integrity of the message with the SAML token[15]:

- Specify the user by assertion
- Verify the sender's authentication contexts with an XML signature
- Acknowledge the sender with the public key that is included in the assertion issued by the third party (IdP).

ID-WSF requires that attributes be stored in the Web Service Provider (WSP) by the user in advance, and then a pointer and the item lists are registered at the Discovery Service (DS) that discovers the URL of the WSP. When a user accesses the Web Service Consumer (WSC) that provides a service, such as the URL of the WSP provided in advance, WSC requires DS to provide the item list of WSP's attributes to the WSC. Once WSC gets the item list of the attributes, WSC requires WSP to provide the actual attributes that are stored in the WSP. When WSP is required to provide attributes by WSC, the WSP should perform either of the two actions below:

- Return the attributes under the predefined policy
- Ask the user that originally owns his/her attributes each time to get his/her consent if the WSP can provide the attributes to the WSC. This mechanism is called Interaction Service (IS), which is one mechanism of the ID-WSF.

This sequence is illustrated in Figure 3. As a technology that manages and operates attributes, ID-WSF has been deployed in many industries, including industries handling digital contents that coordinate information provided by multiple devices and medical science related industries that use a lot of sensitive data. Many investigations and experiments were held in Japan[16,17,18,19,20].

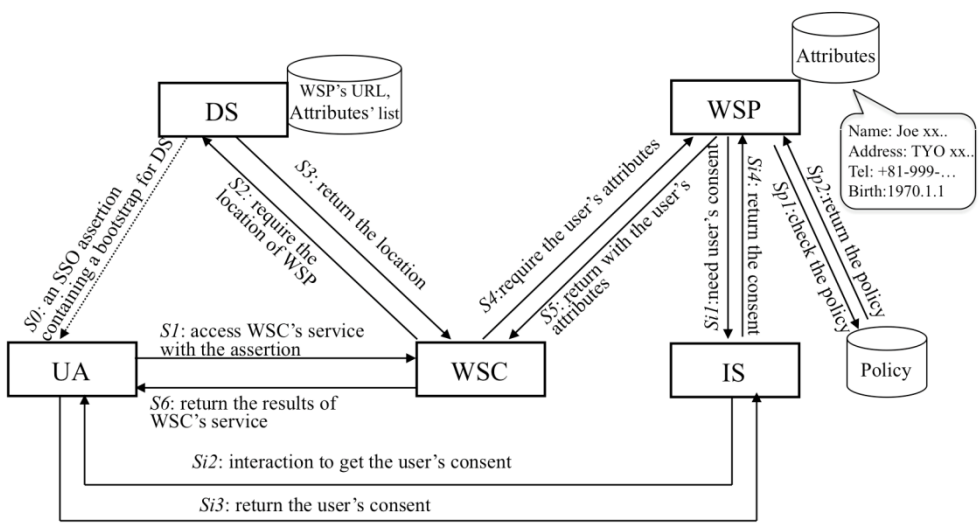

Figure 3. Sequence of the ID-WSF

3.2 Problems of ID-WSF

ID-WSF needs many sites that give mechanisms and repositories, so its sequence is very complicated. Currently, no other web service specifications consider privacy and security for the mechanism and architecture. ID-WSF 1.0 was launched in 2003 and the current version is ID-WSF 2.0, which was extended with additional sub-mechanisms such as People Service (PS). However, if the progress of designing ID-WSF is compared to the progress of cloud computing that gives fast and numerous computer resources for consumer services, the ID-WSF specification is lacking in some areas, including those stated below.

- A user attribute is managed just as a static attribute. The technology and architecture cannot support dynamic attributes that increase every moment of every day.
- If two sites, in particular, the WSC and the WSP, are located in distant places on earth, problems of latency exist for SOAP messaging.
- No standard specifications for logs are defined. Every site stores the log entries to the disk in a local or remote site in an independent format.
- If an auditor wants to assure the system that uses ID-WSF, he or she needs to visit all of the different sites that store each log entry.

4. SOLUTION IN CLOUD COMPUTING ENVIRONMENT

Service providers get personal information prior to permitting their customers to use their services [21]. This means that the user stores his/her static attributes as their identity to one or more specific providers. However, the number of mobile devices such as smart phones is increasing, and the quantity of "active" data such as censor data or GPS information is also increasing. This active data reflects human behavior, which is called the "life log", and are widely distinguished as dynamic attributes.

As described in Section 1, the dynamic attributes of Internet users are stored "somewhere on the net". Historically, web service technology was intended to primarily use static attributes among the vast quantity of information on the Internet. A few technologies access dynamic attributes, such as the APIs for Facebook and Twitter. However, in the scope of federated authentication and identity technology, no investigations of the use of dynamic attributes have considered speed and safety in the cloud computing environment, although some studies have been conducted on the

registration method technology of ID mapping and authentication infrastructure[22, 23, 24]. For these reasons, a new technology is proposed to use both static and dynamic attributes based on extended ID-WSF.

4.1 Separation of Services to Provide Dynamic Attributes

ID-WSF requires the WSC to get the attributes stored in the WSP using SOAP messaging, which is the format defined in the specification of the Data Service Template. It is basically possible to operate a dynamic attribute only if descriptions of the claims, which are written in the Personal Information Profile (PIP), are added. However, each dynamic attribute is added to a repository in a very short time and the quantity is large. As described in the Section 3 problem, it is difficult to operate dynamic attributes in the same way as static attributes, which are not often added or modified. Therefore, the repository and the operating mechanism of dynamic attributes are extracted and separated from the repository and the operating mechanism of static attributes. Listed below are some definitions.

- The WSP that operates static attributes is a WSP.
- The WSP that operates dynamic attributes is a WSPd.
- The WSP owns the repository of static attributes.
- The WSPd owns the repository of dynamic attributes.

The following are three extended features of the ID-WSF as compared with the original ID-WSF:

Extends the PIP to put the URL as a claim that points to the site that operates dynamic attributes

- Adds a mechanism so that the WSC points to the site that operates dynamic attributes
- Adds a Data Service Template to operate dynamic attributes.

The whole architecture is illustrated in Figure 4.

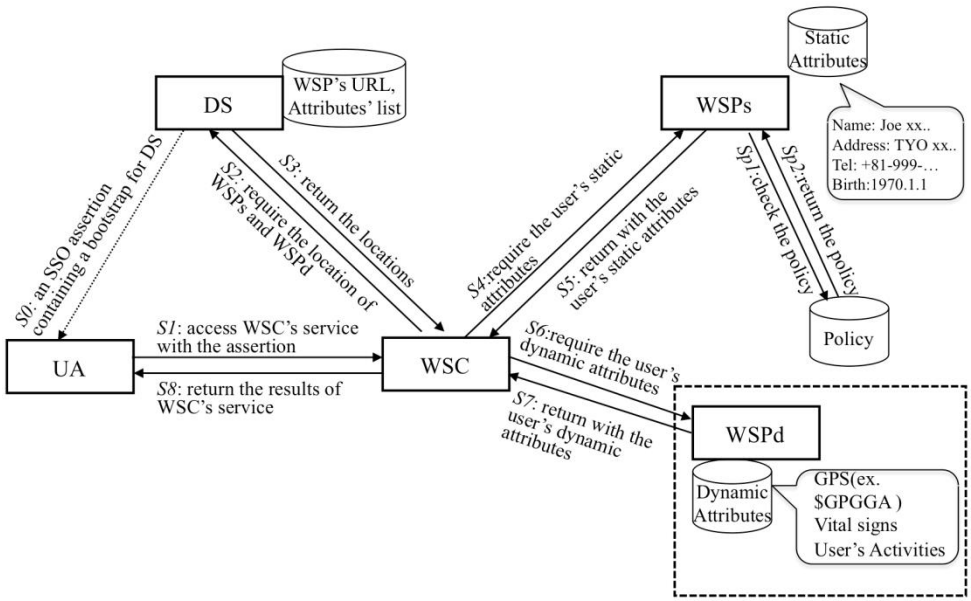

Figure 4. Separated operating mechanism for dynamic and static attributes

4.2 Roaming of Service Function to Provide Dynamic Attributes

As described in Section 4.1, the operating mechanisms of dynamic and static attributes are separate. The WSC must access dynamic attributes in the WSPd through the Internet. This mechanism requires fast network communication between the WSC and the WSPd. In particular, if the WSC and the WSPd continuously exchange attributes with each other through user interactions, the services provided by the WSC to users are highly expected to affect the user experience due to the high latency between the WSC and the WSPd. If the WSC is at a site located with small network distance to the WSPd, it is supposed that network latency is not a problem. At farther distances leading to latency, a roaming mechanism is proposed. With the roaming mechanism, the WSPr, which works with the WSPd, is pushed to a site with small network distance and transfers the dynamic attributes to the repository in the WSPr so that the WSPr alternates with the mechanism of the WSPd. Figure 5 illustrates the change of architecture after deploying the roaming mechanism.

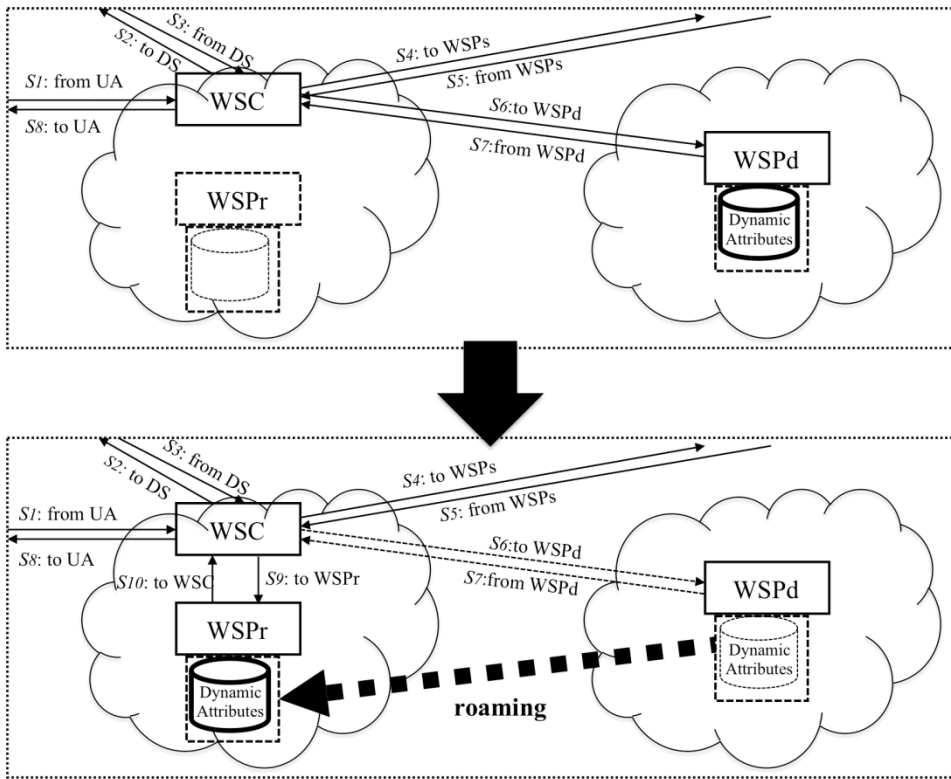

Figure 5. Roaming architecture to help manage dynamic attributes

4.3 Architecture of Secure Logging

A log is necessary if auditors are required to assure that a system runs correctly. Many studies have been done on logs, some of which are about secure logs. Bruce Schneier and John Kelsey describe a computationally cheap method for making all log entries generated prior to the logging machine's compromise impossible for the attacker to read, and also impossible to undetectably modify or destroy[25]. The premise is that log entries are generated at untrusted machines, U_0, $U_1 \ldots U_n$, and sent to the trusted machine *T*. Auditors may perform a task of the information system audit completely and efficiently if they find only the machine *T* that holds all of the log entries at U_0, $U_1 \ldots U_n$. But there is no description of *U* and *T* in the distributed system environment. It is considered that there are further requirements in the cloud computing environment. In particular, distributed web services, which are deployed across multiple domains in the cloud computing environment, are complex and it is very difficult for auditors to find the log entries that are provided by each service site. To solve this problem, the system needs to satisfy the following requirements.

- Centralized logs - The archives of the log entries are aggregated and operated securely at only one site among multiple domains even in the cloud computing environment.
- Secure log messaging - All of the log entries generated by each service site across multiple domains are sent to the archive site securely.

To meet the requirement of "Centralized logs", the formerly proposed architecture that is drawn in Figure 4 can be enhanced to the "Cloud Log Archiver" (CLA) architecture that is drawn in Figure 6. In this architecture, each site, which is DS, WSC, WSPs and WSPd, does not hold any log entries that are locally generated by the service programs running in the site. Instead of writing them on disks, each site sends the log entries to the CLA that holds all of the log entries. The auditor may conduct audits on the CLA only instead of examining each site individually.

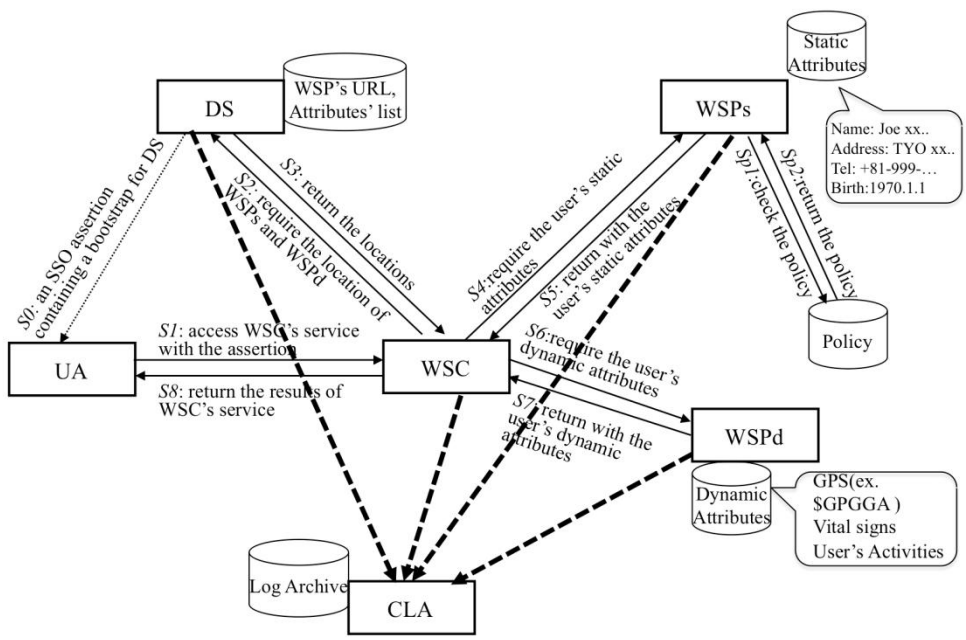

Figure 6. Cloud Log Archiver

To meet the requirement of "Secure log messaging" among multiple domains, it is considered that using SAML assertion is the suitable solution. As described in Section 3.1, ID-WSF represents secure messaging using SAML assertion, which specifies the user, verifies the sender's authentication contexts with an XML signature, and acknowledges the sender with the public key that is included in the assertion issued by the third party (IdP) between two sites. The suggested method of secure log

messaging using SAML assertion, which is issued by DS/IdP to specify the Logger, verify the contexts with an XML signature and acknowledge the logger as the sender, is drawn in Figure 7. Therefore, the Logger is defined as WSC and the Archiver is defined as WSP.

Prior to sending log messages from the Logger (WSC), the Logger requests DS/IdP to send the URI of the Archiver (WSP) and the assertion for the Archiver. Using the assertion, the Archiver verifies the sender's authentication contexts with an XML signature and acknowledges the sender with the public key issued by DS/IdP.

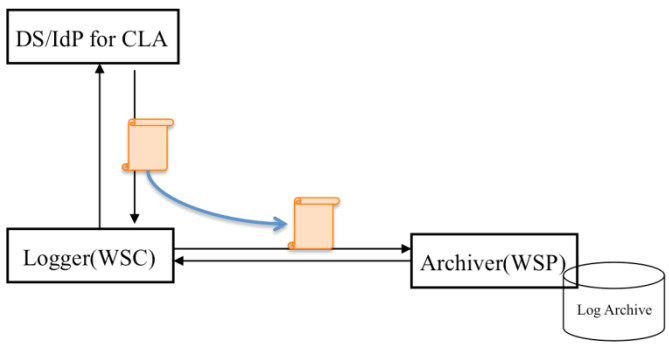

Figure 7. Cloud Log Archiver

The structure of the SOAP message, which is called a SOAP Envelope that consists of a SOAP header and a SOAP body, is drawn in Figure 8. The SAML assertion is described in the element of <saml2:Assertion> in the element of <wsse:Security> in the header of the SOAP Envelope. Another element in <wsse:Security> is the <ds:Signature> that consists of the XML Signature for the SOAP header and the SOAP body. The combination of the SAML assertion and the XML signature for the SOAP header and the SOAP body prevent manipulation and assure secure messaging between the logger and the archiver. The consecutive log records are put in the element of <LogItem> of the SOAP body.

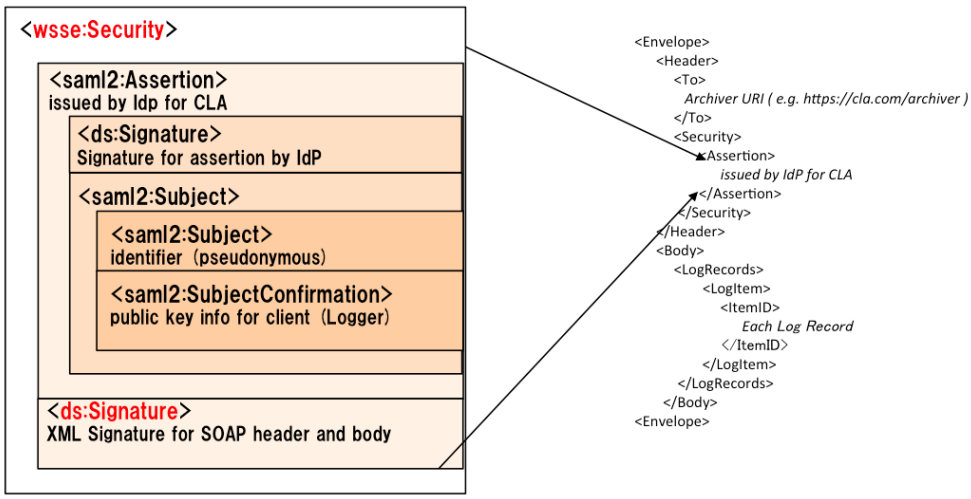

Figure 8. SAML Assertion for CLA

Consequently, the mechanisms of the cloud log archiver correspond with two requirements, which are the centralized logs and secure messaging that assure non-manipulation.

5. EXPERIMENT TO MEASURE LATENCY

The previous section describes the roaming architecture that extends the ID-WSF and separates the operating mechanisms of dynamic and static attributes. In this architecture, dynamic attributes are pushed to another alternative site located with small network distance to the WSC. Meanwhile, as described in Sections 2.1 and 3.2, the latency problems of SOAP messaging are supposed. Therefore, prior to starting implementation, it is desirable to gather basic data for the best implementation of sites that deploy SAML SSO POST and Artifact Bindings in the cloud computing environment. Therefore, the elapsed times of each sequence between multiple SAML IdPs and SPs were successfully measured under real-world conditions.

5.1 SAML SSO in Cloud Computing Environment

Experiments were conducted to measure the performance of federated authentication using SAML to determine whether authentication could be performed quickly enough in a cloud computing environment all over the world. For the experiments, the Amazon Elastic Computing Cloud (EC2) in seven regions of Amazon Web Services (AWS) was selected as the Infrastructure, as a Service (IaaS) for providing virtual instances. In each

region, two sites of each instance, either SAML IdP or SAML SP, were implemented so that the number of sites was 14. The specifications of the AWS EC2 used in the experiments are shown in Table 1. The specifications of the client PC used are shown in Table 2.

Table 1. Specification of Amazon EC2 instance in the experiments

Region	us-east-1(Virginia), us-west-1(Oregon), us-west-2 (California), eu-west-1(Ireland), sa-east-1(Sao Paulo), ap-northeast-1(Japan(Tokyo)), ap-southeast-1(Singapore)
OS/Middleware	Fedora 15 community Edition 32bit / JDK1.6, Tomcat6, OpenAM9.5.3
Instance Type	Small (1.7 GB of memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB of local instance storage)
Misc	Elastic IP for all 14 instances instances

Table 2. Specification of client PC in the experiments

Location	Tokyo
OS	Mac OS X 10.6.8
Web Browser	Firefox 9.0.1
LAN	Wired, up to 100Mbps

5.2 RTT between Web Browser, IdP and SP

One client PC, one IdP in each of the seven regions, and one SP in each of the seven regions were implemented. The number of combinations was 56. For each combination, the RTT was measured. The results are shown in Table 3. The summary of the results is as follows.

- The destination with the maximum RTT from the client PC was Sao Paulo and was almost 30 times longer than the RTT from Tokyo.
- The maximum RTT between the origin/destination among multiple regions was Ireland/Singapore and was 20 times longer than the shortest RTT, which was Oregon/California.
- The RTT within the same region was shorter than 1 msec.

Table 3. RTT between client PC and AWS EC2 regions in the experiments

(Unit: upper: msec, lower: num of hops)

From \ To	Ireland	Sao Paulo	Virginia	Tokyo	Oregon	California	Singapore
Browser	262.257 (25)	315.358 (22)	176.226 (19)	11.294 (11)	124.768 (21)	116.191 (19)	80.260 (14)
Ireland	0.595 (6)	209.370 (17)	94.720 (15)	284.699 (17)	174.144 (24)	157.955 (22)	458.114 (19)
Sao Paulo	208.338 (17)	0.582 (10)	140.327 (18)	314.552 (21)	219.416 (21)	183.778 (20)	351.245 (21)
Virginia	95.516 (18)	150.102 (19)	0.963 (8)	189.889 (18)	98.864 (17)	83.758 (15)	260.546 (17)
Tokyo	267.631 (21)	294.169 (18)	194.126 (19)	0.542 (6)	137.046 (20)	126.557 (18)	82.490 (14)
Oregon	170.743 (22)	224.769 (19)	98.878 (19)	123.098 (21)	0.718 (10)	20.546 (14)	317.168 (23)
California	442.614 (20)	376.727 (20)	248.398 (18)	86.477 (16)	305.802 (19)	300.779 (16)	0.619 (6)
Singapore	442.614 (20)	376.727 (20)	248.398 (18)	86.477 (16)	305.802 (19)	300.779 (16)	0.619 (6)

5.3 Measuring Elapsed Time of SAML SSO on AWS EC2

Under the profile of SAML SP-Initiated SSO: POST/Artifact Bindings, both the IdP and the SP were implemented in each of the seven regions of the AWS EC2. Figure 6 shows the sequence of the profile. The 24 measured points were *t1*−*t24*. Defined below are *T1*, *T2*, and *Tu*:

- *T1*: Duration from the time at the start of accessing the service resources of SP to the time at the finish of displaying the login window
- *T2*: Duration from the time at the completion of the login to the time at the finish of displaying the service resources
- *Tu*: Duration from/to the time at the start/finish of the login by the user.

Therefore, the duration from the time at the start of accessing the service resources of SP to the time at the finish of displaying the service resources is *T*, which is defined below.

- $T = T1 + Tu + T2$

The value of Tu depends on the individual, so the elapsed time for this set of sequence Ts is defined below.

- *Ts = T1 + T2*

Ts is the real response time of the service, so the length of *Ts* affects the user experience.

As shown in Table 4, the results of measuring the elapsed time of the IdP/SP are as follows.

- The maximum *T1* is 989 msec for Ireland/Sao Paulo.
- The maximum *T2* is 1,929 msec for Singapore/Ireland.
- The maximum *Ts* is 2,540 msec for Ireland/Sao Paulo.
- The minimum *T1*, *T2*, and *Ts* are Tokyo/Tokyo.

The maximum *Ts*, which is Ireland/SaoPaulo, is almost 10 times longer than the *Ts* of Tokyo/Tokyo. *T1* is the "waiting time" for use until starting the next action for login after the first action. However, the time is approximately 1 second only.

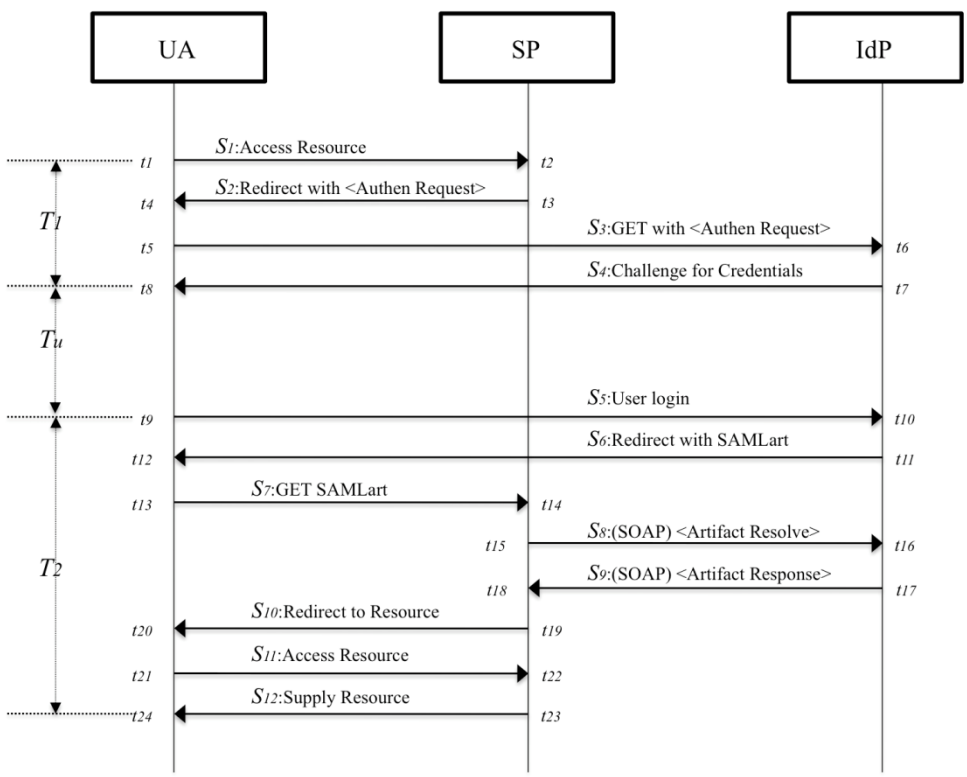

Figure 9 Sequence of SAML SSO POST/Artifact Bindings

Table 4. **Elapsed time between IdP/SP (unit: msec)**

Idp \ SP		Ireland	Sao	Virginia	Tokyo	Oregon	California	Singapore
Ireland	T_1	874	989	838	674	715	426	448
	T_2	955	1,551	1,039	1,042	965	933	1,426
	Ts	1,829	2,540	1,877	1,716	1,680	1,359	1,874
Sao Paulo	T_1	950	702	645	403	902	492	459
	T_2	1,459	1,114	1,267	1,190	1,096	1,121	1,353
	Ts	2,409	1,816	1,912	1,593	1,998	1,613	1,812
Virginia	T_1	655	569	407	229	447	548	539
	T_2	1,003	1,272	640	879	795	702	1,052
	Ts	1,658	1,841	1,047	1,108	1,242	1,250	1,591
Tokyo	T_1	448	395	385	63	173	182	121
	T_2	1,466	1,498	869	199	1,763	683	412
	Ts	1,914	1,893	1,254	262	1,936	865	533
Oregon	T_1	471	509	496	325	468	377	365
	T_2	1,229	1,291	826	560	600	534	1,072
	Ts	1,700	1,800	1,322	885	1,068	911	1,437
California	T_1	523	517	503	190	346	486	237
	T_2	1,066	1,325	799	440	674	660	995
	Ts	1,589	1,842	1,302	630	1,020	1,146	1,232
Singapore	T_1	579	522	318	206	234	267	218
	T_2	1,929	1,595	1,057	442	1,009	1,035	405
	Ts	2,508	2,117	1,375	648	1,243	1,302	623

Table 5. *T1* Elapsed time (unit: msec)

		Singapore / Ireland		Tokyo / Tokyo	
		Elapse	Δ	Elapse	Δ
T_1	t_1	0	0	0	0
	t_2	150	150	7	7
	t_3	152	2	13	6
	t_4	301	149	20	7
	t_5	388	87	30	10
	t_6	430	42	38	8
	t_7	537	107	56	18
	t_8	579	42	63	7

T2 is the elapsed time from the time at the finish of the login until the finish of the data transfer and the start of displaying the web page for the service. The maximum *T2* is approximately 2 seconds, so the user's wait, to see the service page, is 2 seconds.

Table 5 and Table 6 list the differences of elapsed times to compare the *T2* of Ireland/Singapore, which shows the maximum in all combinations, with the *T2* of Tokyo/Tokyo.

For *S8* and *S9* shown in Figure 9, SOAP messaging between the IDP and the SP correspond to *t14−t19*, shown in Table 6. In terms of *Δt*, *Δt15* is 468 msec, *Δt16* is 294 msec, and *Δ*t17 is 3 msec. In terms of TCP/IP packets, *Δt15* is the duration of the sequence [SYN] ⇒ [SYN, ACK] ⇒ [SYN], and it is the time to establish the TCP/IP connection. *Δt16* is the duration of returning [ACK] after the POST. Both *Δt15* and *Δt16* are large.

In contrast, *Δt17* is the residence time in the IdP, which receives the message of *S8*, builds an Artifact Response, and sends the message of *S9*, and is a small value (3 msec).

These experimental results demonstrate that the operation of SOAP messaging, which is generally supposed to require many computer resources and time, has smaller problems than does the problem of latency in the cloud computing environment.

Table 6. *T2* Elapsed time (unit: msec)

		Singapore / Ireland		Tokyo / Tokyo	
		Elapse	*Δ*	Elapse	*Δ*
T2	t_9	0	0	0	0
	t_{10}	41	41	8	8
	t_{11}	58	17	23	15
	t_{12}	131	73	30	7
	t_{13}	136	5	34	4
	t_{14}	286	150	42	8
	t_{15}	754	468	89	47
	t_{16}	1,048	294	89	0
	t_{17}	1,051	3	91	2
	t_{18}	1,230	179	92	1
	t_{19}	1,467	237	121	29
	t_{20}	1,615	148	127	6
	t_{21}	1,629	14	140	13
	t_{22}	1,779	150	188	48
	t_{23}	1,780	1	191	3
	t_{24}	1,929	149	199	8

5.4 Consideration of Latency Having Impacts on Services

Historically, most services on the Internet have had a simple architecture consisting of a web browser and a single service provider. In

Web2.0, which represented "mash-up" services based on REST technology, many service providers provided an API to affiliate with multiple service providers. Due to the expansion and deployment of SSO technology into many fields, the technical infrastructure has been developed to own and operate distributed attributes among multiple sites.

Meanwhile, as described in this paper, the infrastructure has been expected to operate many more quantities of not only static, but also dynamic attributes. So far, each site independently owns the users' attributes and provides services using the attributes, but some sites have recently started providing APIs, such as Facebook and Twitter, to operate these attributes. Corresponding to the increase of sophisticated services, it is supposed that more sites will start operating attributes, and some sites will become "attribute providers" that independently hold and operate dynamic attributes. It is expected that service providers will get many attributes from the attribute providers to provide their specific services on the Internet. In this kind of processing scheme, the RTT between the service provider and the attribute provider is very important.

Due to the expansion of services implemented in the cloud computing environment, it is supposed that the service provider, and the attribute provider are located within network distance to each other. Even in such a network environment with latency, it is expected that the response from the website to the user will be no more than two seconds.

Thus, the mechanism of the dynamic attribute roaming service described in this paper can shorten the RTT and is an effective technology that enhances user experience.

6. CONCLUSION

This paper proposes a new roaming mechanism that separates static and dynamic attributes, extends the SAML/ID-WSF, transfers dynamic attributes to another site located close to the service provider with a small network distance, shortens the RTT and enhances user experience. In addition, in order to assure secure messaging, the cloud log archive mechanism is shown. Due to the successful experimental analysis of large quantities of network latency data for SOAP messaging using SAML SSO in a cloud computing environment, it was shown that the roaming mechanism is expected to effectively improve user experience. In future work, it is necessary to implement the mechanism of roaming in a service system by using the data analyzed in the experiments.

7. REFERENCES

[1] OASIS, *OASIS Security Services (SAML) TC*. Retrieved on March 1, 2005, from http://saml.xml.org/saml-specifications.

[2] OpenID, *OpenID Authentication 2.0 – Final*. Retrieved on December 5, 2007, from http://openid.net/specs/openid-authentication-2_0.html.

[3] Internet Engineering Task Force (IETF), *The OAuth 1.0 Protocol*. Retrieved on April 1, 2010, from http://tools.ietf.org/html/rfc5849.

[4] Internet2, *Shibboleth*. Retrieved on September 10, 2005, from https://wiki.shibboleth.net/confluence/download/attachments/2162702/internet2-mace-shibboleth-arch-protocols-200509.pdf.

[5] Projectliberty, *Liberty Alliance ID-WSF 2.0 Specifications including Errata v1.0 Updates*. Retrieved on March 29, 2008, from http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications/?f=resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications.

[6] SAML Single Sign-On (SSO) Service for Google Apps. Retrieved on August 19, 2013, from https://developers.google.com/google-apps/sso/saml_reference_implementation?hl=us.

[7] Salesforce.com, *Single Sign-On with SAML on Force.com*. Retrieved on June 13, 2014, from http://wiki.developerforce.com/page/Single_Sign-On_with_SAML_on_Force.com.

[8] T. Shimoe, The progress and transition of identity management related technologies. *Japanese Society Artificial Intelligence*, 24(4), p504-511, 2009.

[9] Y. Takeda, S. Kondo, Y. Kitayama, M. Torato, and T. Motegi, Avoidance of performance bottlenecks caused by HTTP redirect in identity management protocols. *Paper presented at the second ACM workshop on Digital identity management - DIM '06*, Alexandria, VA, USA, October 30- November 3, 2006. http://dx.doi.org/10.1145/1179529.1179535.

[10] OASIS, *Security Assertion Markup Language (SAML) V2.0 Technical Overview*, March 25, 2008.

[11] T. Shitamichi, The current status of Cloud Computing and the effort for privacy protection in Europe and United States. *The Law and Computers Association of Japan*, 28, p119-125, 2010.

[12] Forrester Research, *eCommerce Web Site Performance Today - An Updated Look At Consumer Reaction To A Poor Online Shopping Experience*, August 17, 2009.

[13] T. Inoue, H. Asakura, H. Sato, and N. Takahashi, A study of sessions in

the REST architectural style. *Paper presented at the IEICE General Conference 2009*, Japan, March 4, 2009.

[14] T. Ebato, S. Matsumoto, A.Tomono, M.Uehara, and Y. Shimada, The study on implementation authorizing consumer for OAuth. *Paper presented at the 72th National Convention of Information Processing Society of Japan (IPSJ)*, Tokyo, March 9-11, 2010.

[15] J. Sugano, Secure Information Sharing using ID-WSF. Kantara initiative Seminar, Kantara Initaitive Japan Workgroup, June 3, 2011

[16] A. Fujii, K. Ishikawa, T. Morizumi, Y. Kikuchi, T.Yamada, M. Kawamori, and K. Kawazoe, Seamlessviewing service for multi-device users by accession of authentication information. *IEICE technical report*, 108(218), 21-26.

[17] T. Kokogawa, A. Miyajima, H. Ohno, T. Nakamura, and Y. Maeda, A proposal of information delivery platform for medical and healthcare information services. *Information Processing Society of Japan*, 46(14), p1-6, 2009.

[18] K. Horikawa, Development and operation of the ID federation service for consumers and its strategy. 2010 IEICE technical report. Information network. *Paper presented at the Institute of Electronics, Information and Communication Engineers (IEICE)*, Hawaii, USA, January 7-8, 2010.

[19] Y. Chigusa, A. Fujii, K. Ishikawa, Y. Homma, T. Obi, M. Yachida, and Lee Joong Sun, User-device authentication federation framework for receiving personalized telecommunication services based on data broadcasting service. *Forum on Information Technology, Fukuoka, F IT2010*, 9(4), 255-258.

[20] M. Hatakeyama, and S. Shima, Privilege federation between different user profiles for service federation. *Paper presented at the 4th ACM workshop on Digital identity management - DIM '08*, Alexandria, VA, USA, October 27-31, 2008. http://dx.doi.org/10.1145/1456424.1456432.

[21] M. Chiba, K. Urushima, and Y. Maeda, Personal attribute provider: A secure framework for personal attribute exchange on the Internet (Information Systems for Society and Humans). *Information Processing Society of Japan*, 47(3), p676-685, 2006.

[22] M. Hatakeyama, Attribute Exchange using a federation proxy connecting multiple federation protocols. *Paper presented at the 72nd National Convention of IPSJ*, Tokyo, March 9-11, 2010.

[23] G. Washio, and Y. Murasawa, Implementation and verification of authentication platoform for cloud computing based system. *Paper presented at the 73rd National Convention of IPSJ*, Tokyo, March 2-4, 2011.

[24] K. Maki, and G.Washio, A study of ID-Mapping registration

mechanism. *Paper presented at the 73rd National Convention of IPSJ*, Tokyo, March 2-4, 2011.

[25] B. Schneier, and J. Kelsey, Cryptographic support for secure logs on untrusted machines. *Paper presented at the Seventh USENIX Security Symposium*, San Antonio, Texas, January 26-29, 1998.