Towards Unconditional Anonymity: Privacy Enforcement Model in Web Services

Yong Yang[1,2*], Jian Yang[1]
[1]Department of Computing, Macquarie University, Australia
[2]School of Computer, University of Electronic Sci. & Tech. of China, China
{yongyang, jian}@ics.mq.edu.au

Abstract

Privacy in web services is of great importance and a critical requirement for any business and non-business environments. The growth of web services has been accompanied by sharing more and more user personal information with web service providers between diverse and heterogeneous computing systems, which has raised concern about possible malicious or accidental unauthorized abuse of user information. The Security Assertion Markup Language (SAML) architecture is an XML standard for exchanging authentication and authorization data. However privacy preserving in SAML is inadequate for user privacy protection. In this paper, the SAML architecture is extended to address this shortcoming. A privacy enforcement model based on ring signature is presented, which provides unconditional anonymity for web service users. This model enables verification of individuals who belong to a specific group with access right without actually being identified by their IDs or names. Therefore the risk of information leak is reduced. Furthermore, even if the third party is corrupted or the ID correspondence relationship is leaked, the individual remains unrecognizable. Meanwhile most SAML authorization between individual and web services can be done without the presence of the third party, which largely decreases communication overhead and enhances the privacy. Finally, a web services conversation establishment protocol is constructed based on this model, which has been implemented in Java/Tomcat.

1. Introduction

Due to the rapid development of web services and their fast spreading, user personal information shared with web service providers is increasing. It concerns users that the leaked information may be abused and bring trouble or even cause "harm" to their lives. Privacy in web services applications has become a critical and imperative problem. It has attracted great attention from both industry and academia. Legislative acts are even enacted to demand privacy preserving in business, such as the *Financial Modernization Act 1999* [6] in US and *Electronic Transactions Act 1999* [9] in Australia. Essential for privacy preserving in web services is the anonymity between two communicating parties. A useful example is eBay auction. When purchasing some products (such as a house), only valid users registered as "pre-approved" eBay members can bid. On the other hand, bidders' user information needs to be protected since it indicates their preferences or capabilities, as well as breeding collusion between bidders and a seller. In fact, eBay has been optimized for privacy to hide bidders' identities from other people except the seller. However, the bidders' identities also need to be protected from sellers on some occasions. Otherwise, sellers can study users' consumption habits from their bidding histories.

Enforcing privacy in web services poses a number of challenges because it involves users' complex and dynamic privacy requirements that should to be handled. Most recent efforts have been focussing on policy languages. Not only the policies are intricate, but also interactive activities between parties often create unnecessary traffic. Furthermore, the third parties are often devised to help establish trust between users and a service provider. However, whether they are consistently reliable further complicates the issue of privacy preserving, as users are wary of scandal involving the trusted third parties.

In practice, privacy requirement is often integrated with processes of authentication and authorization, which are clearly described by the Security Assertion Markup Language (SAML) [2] standard. This architecture is an XML-encoded framework that defines exchanging authentication, subject attribute and authorization information between online business partners before service invoking. It does benefit business a lot, because of its distinctive features: platform neutrality, loose coupling and improved online expe-

*This work was performed during the author's scientific visit at Department of Computing, Macquarie University, Australia.

978-0-7695-3313-1/08 $25.00 © 2008 IEEE
DOI 10.1109/SERVICES-2.2008.8

rience for end-users.

SAML 2.0 is the most recent OASIS standard, ratified in March 2005 by the OASIS Security Services Technical Committee. Its core specification [1] defines the format of expressing assertions and several protocols for exchanging these assertions. SAML assertions and protocol messages are encoded in XML and then embedded in HTTP/POST requests or XML-encoded SOAP messages for transport. An assertion is a package of information that provides zero or more statements from an issuer. According to the specification, assertions can be generally categorized into the following three types:

- *Authentication*: It indicates that an assertion subject has been authenticated by a particular means. A flexible mechanism is provided to describe the fact the subject was authenticated indeed and how it was done.
- *Attribute*: It defines properties for a subject. Each attribute includes a name (which uniquely identifies this attribute) and a set of values (such as one person could have several phone numbers or e-mail addresses). The assertion subject with the supplied attributes was authenticated by a particular means.
- *Authorization Decision*: It indicates that an access decision has been made to approve or reject a request, by which the assertion subject demanded for a permission to access a specified resource.

Privacy in ongoing web service standards has been considered to some extend. Anonymity protects the privacy of the user's location and true name. In SAML anonymity means not being able to link a sending or receiving to a subject within a set. Actually, SAML 2.0 has a number of features that promote privacy in some way, such as the definition of a pseudonymous identifier by which two service providers can refer to individual principals without revelation of user's identity. Such an opaque string protects the privacy of the user because it inhibits collision between multiple providers (which is possibly caused by a global identifier such as an email address. That kind of information in one domain may indicate user's identity in another domain). Furthermore, two notions are built-in in SAML 2.0 to promote privacy. Persistent identifiers provide a permanent privacy-preserving federation since they remain linked to the local identities until explicitly removed. Transient identifiers support anonymity at a service provider before conversation terminates since they correspond to a "one-time use" identifier created at the site of an identity provider (seen as the third party in some way). However, if the identity provider is corrupted or the ID list is leaked, it is unlikely that the anonymity would be maintained. In other words, SAML only supports anonymity in certain circumstances.

In this paper we investigate an approach to support anonymity under no circumstances, which is referred to as *unconditional anonymity*. In order to support unconditional anonymity, the SAML architecture is extended and the security enforcement model in web services based on ring signature [11] is proposed. Our model adopts *Attribute statement* for the purpose of attaching key-related information to identify subjects. The key-related information in SAML stays consistent with the XML Signature specification [3]. In order to incorporate ring signature into SAML, X509 certificate approach is adopted in this paper. X509 certificate indicates that the subject authenticated by a digital signature where the key was validated as part of an X.509 Public Key Infrastructure. So all we need to concern in our attribute assertion is nearly the *KeyInfo* and *Signature* elements. However, it is worth noting that this model can be implemented in other ways, such as Kerberos. Moreover, for being compatible with SAML 1.1, we choose *SOAP binding*.

To sum up, this paper makes the following main contributions:

- Ring signature is introduced and adapted to achieve unconditional anonymity with regard to privacy in web services. Even if ID information is leaked later on, the user can not be identified. Meanwhile the control of privacy preserving shifts from the third party to users themselves, which greatly increases users' confidence and promotes privacy.
- The SAML architecture is extended to enhance privacy in web services. The user can hide in a specified group. Nobody else can notice or prove his behaviour with respect to service invocation. In addition, after users interacted with an identity provider once, no further activities with it are needed to renew the invocation of web services (providing the IDs is not expired), minimizing communication overhead.
- A privacy enforcement model is exploited in web service conversations, which shows it eminently suitable for the emerging OASIS standards. A secure and privacy enhancement establishment protocol is proposed and implemented under a common web services hosting environment: Java/Tomcat.

The rest of this paper is organized as follows: Section 2 presents the related work. In Section 3, we present the architecture of our SAML extension. In Section 4, we elaborate on our model, mainly analysing the security of this framework. In Section 5, we describe the implementation of our model and propose a privacy preserving conversation establishment protocol based on the model. Finally, in section 6 we summarize our work and provide suggestions for future work, concluding this paper.

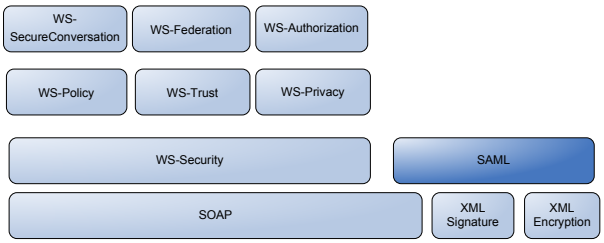

Figure 1. SAML in WS Framework

2. RELATED WORK

Privacy in general has been exploited for years. However, privacy in web services is still under development. Research to date has been focused on developing privacy languages. Rezgui et al. [10] investigate the feasibility and provable reliability of privacy preserving solutions for web service infrastructures. Yee [15] designs privacy controllers together with user privacy policies to protect privacy. Squicciiarini et al. [12] provide a set of assertions to define the privacy related properties. But none of them addresses the issue of enforcing privacy that confirms to emerging industry standards.

The web services security standards, such as WS-Security, WS-Trust, WS-SecureConversation, WS-Federation, WS-Authorization, and WS-Policy, merely provide secure communications between two trusted parties. The basic architecture of WS-framework is illustrated in Figure 1. In particular, the XML and SOAP standards provide a sound basis for SAML.

The SAML is a framework supporting exchange of security information between business partners. Only a few researchers extend its expressive ability, and almost on SAML 1.1. Wang et al. [14] exploit SAML's inherent extensibility to create a delegation framework. Taking account of security features like purposes and obligations among roles, in [8] Ni et al. present privacy-aware access control models. In the follow-up work, [7] describes a conditional privacy-aware role based access control framework to achieve both expressiveness and efficiency. However these proposals focus on different problems. Without consideration of privacy, Wang [13] presents a web services secure conversation establishment protocol based on forward trust, which is similar to our conversation establishment protocol. It can be a complement with privacy enforcement.

In cryptography, ring signature is a type of digital signature that can be performed by any member of a group. Each user can sign a message on behalf of the group with his/her private key. But it can not be determined which of the group members' keys was used to produce the signature. Unlike group signatures, it has two special features: (1) There is no way to revoke the anonymity of an individual signature; (2) Any member can sign the message without additional setup. Nevertheless, the interesting features provided by ring signature have been explored mainly in cryptographic application so far. Au et al. [4] contribute to a combination of the ring signature with identity-based cryptography. Benjumea et al. [5] discuss a general semantic extension of X.509 by way of group signature, ring signature and traceable signature. But it does not consider implementation and integration with industrial standards.

3. SAML PRIVACY ENFORCEMENT

Our SAML privacy enforcement framework consists of three XML-based components: privacy assertions, protocol requests and protocol responses. They all are derived from the corresponding SAML 2.0 schemas. Besides, it can fit into other existing WS-framework standards as well.

3.1. Overview of the privacy enforcement model

A privacy enforcement model is introduced in this section. The basic idea is that a user can remain anonymous in a selection from candidates during the process of service invoking.

Figure 2 illustrates a generic scenario for our SAML privacy enforcement model. The model consists of three roles: a client, a service provider and an identity provider. Suppose a client Bob plans to access web service W. We assume that the identity provider is at least semi-trust, that is, the assertion from it is unfeigned. In order to protect Bob's identity for some reasons, Bob demands to hide himself among a group of people who have similar access rights, which can be verified by service providers. In our model, he can first request an assertion from the identity provider. Then the identity provider responds to him with a list of candidates, from which he can randomly choose some as a group to hide himself in. During invocation, Bob sends the invocation message, the attached ring signature, and the assertion to service W. Lastly, after verification, W can confirm that the client is from the specified group but Bob's identity can be revealed under no circumstances. In this way Bob's personal information is completely protected.

If Bob renews this invocation, he directly reuses the assertion before its expiration. This can largely reduce both communication and computing overhead for the identity provider, which is supposed to be a bottleneck in many occasions.

3.2. Privacy enforcement Assertion

In this section, we shall investigate the details in our model. Let $s_1, s_2, \ldots, s_n$ be a group set $S(n > 1)$, ran-

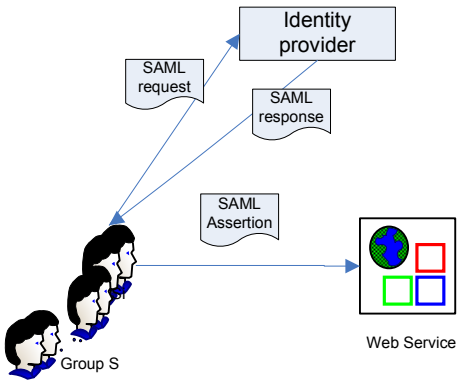

Figure 2. Privacy enforcement model

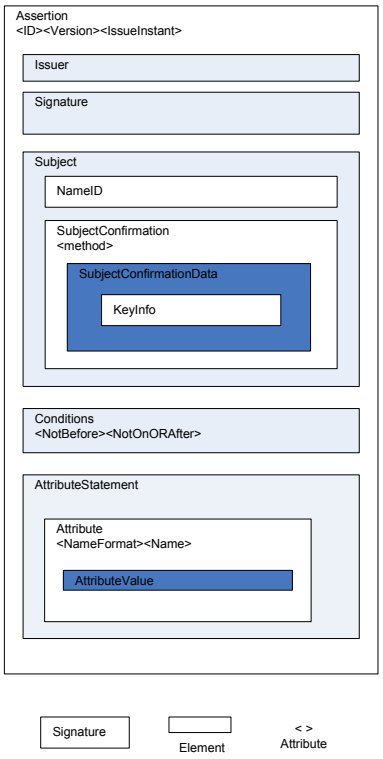

Figure 3. Privacy enforcement assertion structure

domly selected by an actual invoker from the list of candidates returned by the identity provider. Without loss of generality, $s_j (1 \leq j \leq n)$ is assumed to be the ID who actually invokes service W anonymously, such as Bob mentioned in the preceding paragraphs. The privacy enforcement assertion can be expressed as *attributeStatment* in Figure 3.

- **Issuer**: It is a string to carry the issuer's name.
- **Signature**: It represents the signature of the issuer for the assertion.
- **NameID**: This element describes the identifier for the subject. In our model, it represents the group S, which is selected from candidates. It is important to choose individuals carefully to promote privacy. For example, make sure the members of group S within their lifespan during the period of invoking. We can use colon (:) marks to concatenate all the identifiers of individuals in the group S. For instance, if such group includes three persons: Alice, Bob and Lily, the NameID should be "Alice:Bob:Lily".
- **SubjectConfirmation**: This element provides the means for web services to verify the correspondence of the subject and the party it is communicating. Two kinds of most useful ways: *holder-of-key* and *sender-vouches* are provided by SAML. Holder-of-key requires the attesting entity to present an XML-based signature that can be verified by the KeyInfo information included by SubjectConfirmationData. Whereas send-vouches requires web service to verify the attesting entity based on former trust relationship. *Holder-of-key* is preferable because of privacy concern. Besides, it is more flexible because previous trust relationship is not a prerequisite.
- **KeyInfo**: It represents cryptographic keys that are used to authenticate an attesting entity. The X509 certificate has been designed to build a public key to a subject. Such a subject is only one that knows the associated private key.
- **Conditions**: Conditions must be evaluated when assessing the validity of the assertion. *NotBefore* and *NotOnOrAfter*, together with *IssueInstant* define the exact lifetime of the assertion.
- **AttributeSatement**: It asserts a multi-valued attribute associated with the authenticated principal. In the response assertion, all the group public keys information is linked by colon (:) marks with each other in the same order of NameID element. For instance, the attributevalue for Alice, Bob and Lily may be "XD6s...:ZCCA...:ors...". In addition, the correspondent life expectancy is further supplied to assure the validity of each individual.

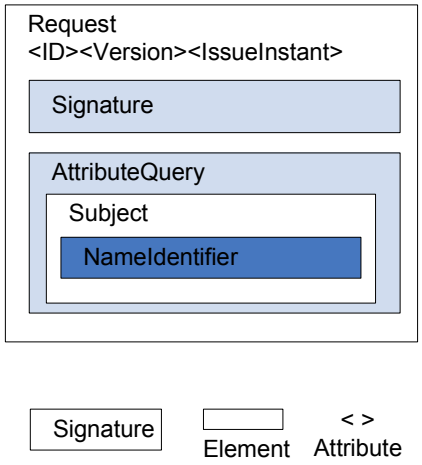

Figure 4. Privacy enforcement request structure

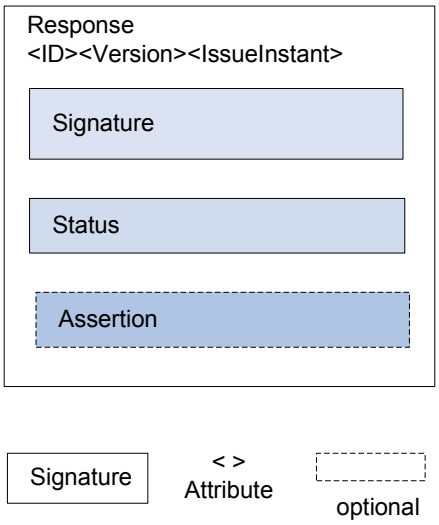

Figure 5. Privacy enforcement response structure

3.3. Privacy enforcement Request & Response

The privacy enforcement request and response messages conform to the SAML request and response protocols. The procedure of interactions is described as follows:

1. The user s_j sends a request to the identity provider for a list of candidates. The request includes some form of user's identity and information about target service W. Its structure is shown in Figure 4.
2. The identity provider authenticates the request. If s_j is accepted, the identity provider responds with candidate list to s_j. The response structure is illustrated in Figure 5. Otherwise, this interaction fails and returns to step 1.
3. s_j verifies the assertion by signature element. If the verification is successful, s_j creates ring signature for the invocation message according to following algorithm 1 and sends the message, the attached ring signature, as well as the assertion to service W, or otherwise returns to step 1.

Algorithm 1: Create Ring Signature

Input: assertion, message m.

Output: ring signature σ.

(a). Extract information about candidates' IDs, lifespan, as well as their public keys from the assertion. Select several candidates as a group S to hide s_j during invocation. Meanwhile, make sure all the selected candidates will not be expiry at that time.

(b). Consider a security parameter b. Get a random number glue value $v \in \{0,1\}^b$ from unique serialized number sequence as a timestamp. Compute the symmetric key k. h denotes a common public collision-resistant hash function, such as SHA-1.

$$k = h(m) \tag{1}$$

(c). Pick n-1 random numbers as challenge values x_i for all members in the group S except s_j. $g_i(x)$ denotes a one-way trap-door function (A one-way function means it is easy to compute but difficult to invert. Only someone who knows secret parameters, such as a private key, can invert that function efficiently). Compute correspondent response values.

$$y_i = g_i(x_i)(i = 1, \ldots, n, i \neq j) \tag{2}$$

(d). Solve for y_j, where $\bigoplus$ represents the exclusive-or operation on bits and E_k represents the symmetric encryption with key k.

$$C_{k,v}(y_1, y_2, \ldots, y_n) = E_k(y_n \bigoplus E_k(y_{n-1} \bigoplus E_k(y_{n-2} \bigoplus E_k(\ldots \bigoplus E_k(y_1 \bigoplus v)\ldots)))) = v \tag{3}$$

(e). Invert the x_j,

$$x_j = g_j^{-1}(y_j) \tag{4}$$

(f). Output σ

$$\sigma = (v, x_1, x_2, \ldots, x_n) \tag{5}$$

4. After receiving the invocation message, the assertion and the ring signature from the client, service W verifies the signature. The algorithm of verification is presented in the following:

Algorithm 2: Verify the ring signature

Input: ring signature σ, assertion, message m

Output: accept or reject: Boolean

(a). Verify the assertion, extract the public-key information. Check each group member's lifespan to ensure the validity.

(b). For $i = 1, 2, \ldots, n$ compute response values correspond to x_i in the ring signature.

$$y_i = g_i(x_i) \tag{6}$$

(c). Compute the symmetric key k

$$k = h(m) \tag{7}$$

(d). Verify the ring signature equation.

$$C_{k,v}(y_1, y_2, \ldots, y_n) = v \tag{8}$$

Only if all the verifications are passed, can W confirm the client is indeed from such a group.

These algorithms, which are based on ring signature in cryptography, have been adapted to web service environment. Intuitively, ring signature has no indication or reflection to s_j. So nobody can identify the individual who have signed this signature even if the private keys are leaked, that is, the actual user could be identified in no circumstances.

4. Analysis

The identity of Bob is unconditionally protected with our model. To see this, we rethink our goal first. Anonymity is defined as the state of being indistinguishable within a group, namely the anonymity set. Conditional anonymity requires that a user can remain anonymous until some conditions are violated. That is to say, with the assistance of third parties the user's identity can be revealed in some occasions. In contrast, unconditional anonymity emphasizes Bob's identity can be revealed under no circumstances. According to our model, service W could judge the user only from ring signature and the assertion. But the assertion merely indicates the identities of the group instead of individuals. So our remaining task is to prove unconditional anonymity in the signature schema.

The idea of a ring signature is that any user can sign a message on behalf of a group of users, including the user himself/herself. We assume Mallory is an attacker with the ability to intercept or eavesdrop on messages during the process of service invoking. Additionally, it is supposed that every member's private key in the group will not be exposed during the communication. So Mallory can not obtain any member's personal private key in our model. The possible methods of attacks are discussed in the following:

- If Mallory can intercept the ring signature defined in Eq. 5, for each challenge value x_i (The outcome x_j in Eq. 4 is also unpredictable when inverting the one-way function) and glue value v is of randomness, Mallory can not indicate the Bob's identity.
- Even if Mallory decodes public keys of S from the assertion, he could barely forge ring signatures. The main difficulty lies at the $C_{k,v}$ Eq. 3. To overcome it, we note that Mallory must invert the trapdoor one-way function to get the challenge value x_i from the given response value y_i, or else the equation can not be satisfied. However private keys are unknown to Mallory and the hash function h is collision-resistant, so he can not solve the equation. Conversely, anyone in the group can produce the equation using his/her private key. That is to say, only group member can sign the message.
- Furthermore, if ID information is leaked, nobody can distinguish the actual signer either. Because the signature doesn't imply or deduct signer's private key, Bob can deny it wisely as anyone in specified group could sign it in like manner.

A formal cryptographical proof can be found in [11]. Summing up, people only in the group defined in the assertion are entitled to sign, in order to invoke the target service. But on the other hand service providers could hardly distinguish the actual signer by signatures and assertions. The user's privacy is protected while invoking web services.

Meanwhile, ID information authenticated in assertions remains valid until expiry. So the information of public keys about group S can be reused instead of requesting the identity provider every time, reducing the communication cost largely as well as promoting privacy. Due to dynamics of web services, it is advisable to be scrupulous in choosing candidates to avoid violating anonymity.

5. Implementation

We have implemented the SAML privacy enforcement model on Java/Tomcat. The assertion format is constructed. Additionally, a secure and privacy enforcement conversation establishment protocol is formalized. Figure 6 portraits the framework of our implementation. We can note that the

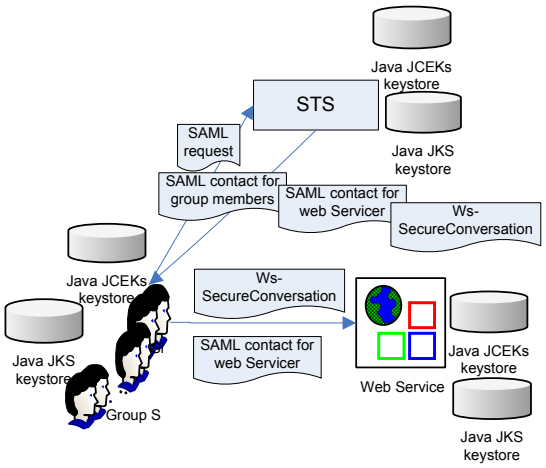

Figure 6. Implementation design

WS-Trust, WS-SecureConversation, etc. can be easily integrated into our model.

Security Token Service (STS) acts as an identity provider in this scenario. The web services conversation establishment protocol is described in the following:

1. Bob sends a request (including information about his identity and target conversation service) to STS for a candidate list. So that Bob will be able to hide himself among the candidates.

2. STS authenticates the request from Java JKS Keystore. If the verification is successful, STS creates WS-SecureConversation Secure Context Tokens (SCTs), including SCT identifier (which involves a conversation key), SCT for group S (since the group is chosen by Bob from candidates, SCT for all candidates actually includes SCT for group S) and SCT for web service W. Java JCKEs keystore is used to store temporary SCTs to speed up the process. Additionally, each SCT token is encrypted using receiver's public key and authenticated by the signature of STS. Afterwards, STS responds with all the SCTs to Bob. Otherwise, the conversation fails to establish and protocol terminates.

3. Bob verifies the SCT identifier and SCT for him, which bind the trust relationship between group S and web service W. If the verification is successful, Bob parses the SCT for him and extracts the conversation key from the SCT identifier, or else the conversation establishment fails and protocol terminates.

4. Bob checks candidates' validity and ensure their IDs will not be expired during invoking. If the verification is successful, he creates ring signature σ. When he begins a conversation with W, he encrypts the message with the conversation key and sends it, as well as the attached ring signature, SCT identifier, SCT for W to service W. Otherwise, the protocol terminates.

5. After receiving the messages, W checks the ring signature and SCTs. If the verification is successful, W can decrypt the messages using the conversation key extracted from W's SCT. That is, the conversation key is established successfully and protocol finishes. Otherwise, the conversation establishment fails and protocol terminates.

```
<?xml version="1.0" encoding="UTF-8"?>
<saml:Assertion
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="2431323453452"
  IssueInstant="2008-02-29T12:10:00"
  Version="2.0">
  <saml:Issuer>http://abcAttributeService.au</saml:Issuer>
  <ds:Signature>
    ...
  </ds:Signature>
 <!--Here is the subject-->
  <saml:Subject>
    <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
     Alice:Bob:Lily
    </saml:NameID>
  </saml:Subject>
  <saml:Conditions>
     NotBefore="2008-02-29T12:00:00Z"
     NotOnOrAfter="2008-05-30T12:15:00Z"
  </saml:Conditions>
  <saml:AttributeStatement>
<!--Here is the public key information-->
    <saml:Attribute
     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
     Name="urn:oid:0.9.2342.19200300.100.1.1">
     <saml:AttributeValue
       xsi:type="xs:string">
        RCV2R...:BC92D...:JSUFS...
     </saml:AttributeValue>
     <saml:AttributeValue type="xs:string">
      20080204120000:
      20080205140000:20080302150000</saml:AttributeValue>
     <saml:AttributeValue type="xs:string">
      20080604120000:
      20080705140000:20080802150000</saml:AttributeValue>
    </saml:Attribute>
   </saml:AttributeStatement>
</saml:Assertion>
```

Figure 7. Example of assertion

All the messages are authenticated by signatures to assure integrity. Moreover, messages from Bob to W are encrypted by the secret conversation key. So attackers can not construct the same security tokens. Meanwhile W can merely ensure the client is from a group, namely anonymity is achieved.

A sample of an assertion is described in Figure 7. Besides, we have used *tcptrace* to record soap messages during invocation, which can be found online at http://www.ics.mq.edu.au/∼yongyang/Anonymity.htm .

6. Conclusion

We have introduced ring signature into web services and extended the SAML standard to achieve privacy enforcement. To our knowledge, this paper has proposed the first lightweight unconditional anonymity model based on SAML specification. None of any third parties can identify the individual even if any ID correspondence relationship is leaked. The user's personal information is completely protected. Another advantage of our model is that most SAML authorization between individuals and web services can be done without the presence of the third party, both largely increasing user confidence and decreasing communication overhead. Finally, we have proposed a conversation established protocol for implementation in web services. It is believed that the protocol is of great practical significance for privacy preserving in web services.

As future work, we plan to exploit privacy enforcement for other WS-Framework using ring signature. In addition, enforcing anonymity on users' behaviours still needs to be exploited. An integrated and complete solution to anonymity enforcement framework is currently under investigation.

References

[1] *Assertions and Protocols for the OASIS Security Assertion Markup Language(SAML) V2.0*. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[2] *Security Assertion Markup Language*. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.

[3] *XML Signature*. http://www.w3.org/TR/xmldsig-core.

[4] M. H. Au, J. K. Liu, T. H. Yuen, and D. S. Wong. Id-based ring signature scheme secure in the standard model. In H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, and S. ichi Kawamura, editors, *IWSEC*, volume 4266 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2006.

[5] V. Benjumea, S. G. Choi, J. Lopez, and M. Yung. Anonymity 2.0 - x.509 extensions supporting privacy-friendly authentication. In F. Bao, S. Ling, T. Okamoto, H. Wang, and C. Xing, editors, *CANS*, volume 4856 of *Lecture Notes in Computer Science*, pages 265–281. Springer, 2007.

[6] Information regarding the gramm-leach-bliley act of 1999, http://banking.senate.gov/conf/. *U.S.Senate Committee on Banking,Housing ,and Urban Affairs*.

[7] Q. Ni, D. Lin, E. Bertino, and J. Lobo. Conditional privacy-aware role based access control. In *ESORICS '07: Proceedings of the 12th European Symposium On Research In Computer Security*, pages 72–89. Springer, 2007.

[8] Q. Ni, A. Trombetta, E. Bertino, and J. Lobo. Privacy-aware role based access control. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 41–50, New York, NY, USA, 2007. ACM Press.

[9] The Parliament of the Commonwealth of Australia, http://www.aph.gov.au/parlinfo/billsnet/99131.pdf. *A Bill for an Act to facilitate electronic transactions, and for other purposes*.

[10] A. Rezgui, M. Ouzzani, A. Bouguettaya, and B. Medjahed. Preserving privacy in web services. In *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*, pages 56–62, New York, NY, USA, 2002. ACM.

[11] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[12] A. C. Squicciarini, A. A. Hintoglu, E. Bertino, and Y. Saygin. A privacy preserving assertion based policy language for federation systems. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 51–60, New York, NY, USA, 2007. ACM.

[13] J. Wang. A web services secure conversation establishment protocol based on forwarded trust. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services (ICWS'06)*, pages 569–576, Washington, DC, USA, 2006. IEEE Computer Society.

[14] J. Wang, D. D. Vecchio, and M. Humphrey. Extending the security assertion markup language to support delegation for web services and grid services. In *ICWS '05: Proceedings of the IEEE International Conference on Web Services (ICWS'05)*, pages 67–74, Washington, DC, USA, 2005. IEEE Computer Society.

[15] G. O. M. Yee. A privacy controller approach for privacy protection in web services. In *SWS '07: Proceedings of the 2007 ACM workshop on Secure web services*, pages 44–51, New York, NY, USA, 2007. ACM.