Mobile Agent and Web Service Integration Security Architecture

Junqi Zhang, Yan Wang and Vijay Varadharajan
Department of Computing, Macquarie University
Sydney, Australia
{janson,yanwang,vijay}@ics.mq.edu.au

Abstract

Mobile agent technology and Web Service technology compensate each other and play very important roles in e-service applications. The mechanism of Web Services technology naturally provides a platform for deploying mobile agent technology. Therefore, the integration of the mobile agent technology and Web Service technology has been actively investigated in recent years. On the other hand, the security issues of the integration system have not drawn much attention. In this paper, we present a new security architecture for the integration of mobile agent and the Web Services technology. This architecture provides a new authentication scheme for Web service provider to verify the mobile agent owner's identity by employing an identity-based signature protocol without using the username/password pair, which is infeasible for mobile agent. We also propose a new Web services and mobile agent system confidentiality protocol, which provides an alternative method to current security mechanisms without using Certification Authorities (CA) based public key infrastructure. With this scheme, it can simplify the key management and reduce the computation load particularly for group-oriented web services. In addition, this scheme also inherently has the non-repudiation property.

1. Introduction

Web services technology is now becoming one of the most important technologies for e-service applications. In a wireless network and pervasive computing enabled environment, the customer can use PDA (Personal Data Assistant) access the Internet to accomplish complex transactions online with other service providers. New PDAs have emerged with more powerful functionalities combining telephone, pocket PC, Wi-Fi network technology (wireless network), GPS (Global Positioning System), Bluetooth technology, and 3G technology. Some operating systems, such as Palm OS and Microsoft Windows Mobile are providing more and more functionalities, and better multimedia and computation facilities. With the development of technologies, handheld devices (PDAs) are becoming more popular, more powerful and less expensive. However, handheld devices are powered by batteries. Their CPU speed and storage space are not comparable to a laptop or a desktop. These constraints require light-weighted, autonomous and efficient systems that consume fewer resources and require less network connection.

Mobile agent is a mobile executable object that can be dispatched by its owner or agent service provider, and migrate in the network autonomously to accomplish the tasks specified by its owner. This new technique can help enable the autonomy and improve the efficiency of the processing in a distributed environment as the dispatched agent can benefit from executing locally. After having accomplished the task, the agent can migrate back to the owner with collected data. During this period, it does not require constant network connection as traditional RPC (Remote Procedure Call) so it is applicable to devices with limited bandwidth and computing resources (e.g. PDA), and long-term transactions without constant interactions [17, 18]. Web Services technology emerged in recent years, which is based on XML (eXtensible Markup Language) SOAP (Simple Object Access Protocol) and WSDL (Web Service Description Language) protocols [12]. This technology enables the communication between software entities and the client and can invoke the services from the service provider to accomplish some tasks, such as information retrieval, online calculation, ticket/room reservation and payment. The grain of a service can be as small as a remote function.

Therefore, Mobile Agent technology and Web Service technology are different but they compensate each other and play the very important roles in e-commerce applications. The mechanism of Web Services technology naturally provides a mechanism for mobile agent technology in several aspects. First, the XML based protocols aim at enabling the communication between software entities, which can be mobile agents and stationary agents. Second, the mobile agent service can be described as a type of the provided services by the service provider. The clients request

IEEE International Conference on Service-Oriented Computing and Applications(SOCA'07)
0-7695-2861-9/07 $20.00 © 2007 IEEE

can be mapped to the mobile agent platform. In addition, to respond to the request, a set of mobile agents can be employed to enable the parallel processing if necessary so as to ensure the efficiency.

As a result, the applications combining of mobile agents and Web service technology have drawn much attention in recent years [19, 16, 8, 3, 7]. Dominic Cooney et al. presented a model for implementing Web services with mobile agents [3]. Fuyuki Ishikawa et al. proposed a general framework for "Mobile Web Services" in [8]. Riccardo Pascotto presented the ACTS AMASE project which described a mobile agent approach for users' access to network-based services [15]. Jan Peters introduced an integration architecture of mobile agents and Web services [16]. Wassam Zahreddine et al. presented an agent-based approach for composite mobile Web services over mobile devices [19]. Some combination schemes of agents and Web services are also presented by other studies [7, 6, 11, 10, 14].

On the other hand, the mobile agent and Web services security is still of a big concern for applications. Currently, Web services and mobile agent security is mostly based on Certification Authorities (CA) based public key infrastructure. Consequently, a trusted third party is required in advance; users must have their key pairs and the service owner must manage all the users' public key and encrypt data using different keys. In some applications, this may be not necessary such as services just for particular groups.

In this paper, we present a new mobile agent and Web services security architecture. This security architecture employs a novel Identity-based public key system and provides a new authentication protocol without using the username/password pair, which is infeasible for mobile agents, and gives an alternative method to current security mechanism without using the Certification Authorities (CA) based public key infrastructure. It simplifies the key management and reduces the computation cost particularly for group-oriented web services.

The rest of this paper is organized as follows. In Section 2, we introduce security issues in Web services and mobile agent system. Section 3 presents our new Web services and mobile agents oriented authentication and confidentiality security architecture. Section 4 presents a simple application example of this security architecture. Finally, some concluding remarks are provided in Section 5.

2. Security Issues in Web Services and Mobile Agent System

Same with the information security, security in mobile agent and web services system can also be divided into several logical component: confidentiality, authentication, authorization, integrity, nonrepudiation, privacy, and availability [13, 4, 12]. Information in a networked system is either in storage or in transit. The term confidentiality is also used to refer to the requirement for the data in transmit between two communication parties (e.g. mobile agent user and the Web services provider) not to be available to third parties. There are two approaches to the confidentiality. One is to use the private connection between two parties. It can be a dedicated line or a virtual private network (VPN). Another approach is to use encryption when information is being sent over an untrusted network. Encryption can be used in three types: symmetric encryption, asymmetric encryption and hybrid encryption. Symmetric key algorithm can use DES (Data Encryption Standard) or its replaced AES (Advanced Encryption Standard). In the mobile agent and Web service system, this algorithm can not be used because the one-and-only key is used and the mobile agent can not take any secret key with it. Both asymmetric encryption and hybrid encryption are based on the public key infrastructure (PKI). It consists of the certificates of different parties issued by Certification Authorities (CA), a repository for retrieving certificates, a method of revoking certificates, and a method of evaluating a chain of certificates from public keys that are known and trusted in advance to the target name [9]. The figure 1 shows how PKI is used for a Web service transaction. It works as follows: Firstly, key pairs are generated for the user and Web service provider and public keys are registered with the registration authority. The certificate authority (Trusted Third party or "TTP") issues a digital certificate with the public key. XML Key Management Specification (XKMS) can be used to provide the PKI service over the Web. Secondly, the Web services user can retrieve the public key of the Web services provider from a PKI directory using Lightweight Directory Access Protocol (LDAP) or Directory Services Markup Language (DSML). Finally, the Web services provide can communicate securely with key pair. The XML signature can be used for integrity and non-repudiation. For mobile agent confidentiality, both agent and the carried information are needed to be secured.

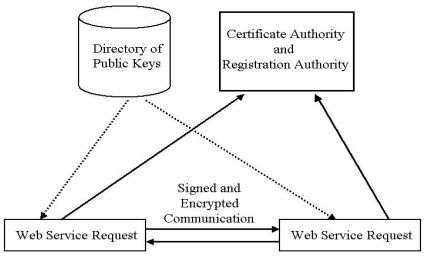

Figure 1. PKI used for Web Services

Integration means that if information is tampered with, this tampering can be detected. Data integration relies on

hashing algorithm and digital signature. Hashing algorithm such as SHA-1, can be used to take a message as input and produce a message digest as output. Then this digest must be encrypted to ensure the integrity.

Non-repudiation means that it can be verified that the sender and the recipient were the parties who claimed to send or receive the message, respectively. In other words, non-repudiation of origin proves that data has been sent, and non-repudiation of delivery proves that it has been received. In mobile agent and Web services system, it means that both an agent and server cannot deny that a given communication exchange has taken place. Digital certificate is a solution for non-repudiation. Digital certification includes the public key, the corresponding private key holder identity information, a serial number, and expiry date. Digital certificate is part of the public key infrastructure.

Authentication is the process of verifying the digital identity of the sender. Authentication process includes PKI technology, Biometrics, password, and hardware. The common feature in these methods is that a token is in the possession of the entity. The token can be hardware-based, such as smartcard or software-based, such as a private key on a hard drive. Web services run on behalf of a human user, a human-to-machine authentication technique (for example a password) can be used for authentication. The WS-Security Addendum describes a means to use a username and password as security token in a SOAP message. The mobile agent authentication invovles the agent owner, agent itself, the server host. In the integration system of mobile agent and Web services, the software-based token cannot be used because the agent can not take the secret key. In the next section, we will propose an identity based protocol for authentication.

Authorization is used to decide if person, program or device X is allowed to have access to data, functionality or service Y. We can say that authentication is about "who you are" and authorization is about "what you are allowed to do". Authorization uses the security policy, which includes the traditional access control list (ACL) and the more popular role-based access control (RBAC). RBAC maps the security to an organization's structure. With RBAC, each user's privileges can be assigned to one to several service groups. In the next section, we will use RBAC for mapping the users and services. In our scheme, each user is assigned to one to many groups and each group correspond to one service.

However, the above security architecture has its drawbacks. Namely, all the users must have his/her public/private key pair based on PKI, and the service server must verify and manage all users' public keys. In addition, the service server has to search the user's public key and use different keys to encrypt messages for different users whenever they send the message to the user. Furthermore, the username/password tokens are required in order to do the authentication. However, in the mobile agent system, the mobile agent could not take any password or private key with them for security reasons while roaming in the networks. This has been a big challenge for mobile agent security research community. In the following section, we propose a new mobile agent and Web services security architecture. In this security architecture, we do not need Certification Authority (CA) based public key infrastructure. Instead, we adopt the ID-based authentication scheme so that the mobile agent is not required to carry a password for authentication. Furthermore, the server only needs one key to encrypt one service that can be available to a group of users.

3. Mobile Agent and Web Services Integration Security Architecture

In this section, we begin with the introduction of the Bilinear Pairings and Bilinear Diffie-Hellman (BDH) assumption. The security of our new schemes are based on Computational Diffie-Hellman Problem and the BDH assumption. Then we describe the integration system of mobile agent and Web services and present the mobile agent and Web services authentication scheme and key distribution and data encryption security architecture. These schemes are based on [1, 5, 2]. Finally, we discuss the scenarios in which the users change group and the Web services change.

3.1. Bilinear Pairings and BDH Assumption

We set up our systems using bilinear pairings. Let us define two cyclic groups $\mathbb{G}_1, \mathbb{G}_2$. $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ is a multiplicative group. Both have a prime order q. Let e be a computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. For $a, b \in \mathbb{Z}_q$ and $P, Q \in \mathbb{G}_1$, we have $e(aP, bQ) = e(P, Q)^{ab}$.

Definition 1 *(Decisional Diffie-Hellman Problem) Given* $P, aP, bP, cP \in \mathbb{G}_1$ *and* $a, b, c \in \mathbb{Z}_q$*, decide whether* $c = ab \in \mathbb{Z}_q$.

A decisional Diffie-Hellman problem (DDHP) is satisfied since $e(aP, bP) = e(P, P)^{ab}$. The security of the pairing algorithm is based on the computational Diffie-Hellman problem (CDHP), which is given below.

Definition 2 *(Computational Diffie-Hellman Problem) Let* a, b *be chosen from* $\mathbb{Z}_q$ *at random and* P *be a generator chosen from* $\mathbb{G}_1$ *at random. Given* (P, aP, bP)*, compute* $abP \in \mathbb{G}_1$.

$\mathbb{G}_1$ is referred to as a Gap Diffie-Hellman (GDH) group and CDHP is referred to as a Gap Diffie-Hellman problem – if DDHP can be solved in polynomial time and no polynomial algorithm can solve CDHP with non-negligible advantage within polynomial time.

Definition 3 *(Bilinear Diffie-Hellman Problem) Given (P, aP, bP,cP) for some* $a, b, c \in Z_q^*$ *to compute* $e(P, P)^{abc}$*, an*

algorithm $\mathcal{A}$ has advantage ε in solving BDH in $(\mathbb{G}_1, \mathbb{G}_2, e)$ if

$$Pr[\mathcal{A}(P, aP, bP, cP) = e(P, P)^{abc}] \geq \varepsilon \qquad (1)$$

Where the probability is the over random choice of (a, b, c) in $\mathbb{Z}_q^*$, the random choice of $P \in \mathbb{G}_1^*$, and the random bits of $\mathcal{A}$.

Definition 4 *(BDH Parameter Generator) A randomized algorithm $\mathcal{IG}$ is a BDH parameter generator if (1) $\mathcal{IG}$ takes a security parameter $0 < k \in \mathbb{Z}$, (2) $\mathcal{IG}$ runs in polynomial time in k, and (3) $\mathcal{IG}$ outputs the description of two groups $\mathbb{G}_1, \mathbb{G}_2$ and the description of a bilinear map e.*

Definition 5 *(BDH Assumption): An algorithm $\mathcal{A}$ has advantage $\varepsilon(k)$ in solving the BDH problem for $\mathcal{IG}$ if for sufficiently large k*

$$Adv(\mathcal{A}) = Pr[\mathcal{A}(G_1, G_2, e, P, aP, bP, cP) \qquad (2)$$
$$= e(P, P)^{abc}] \geq \varepsilon(k)$$

We say that $\mathcal{IG}$ satisfies the BDH assumption if any randomized polynomial time algorithm $\mathcal{A}$ solves the BDH problem with advantage at most $1/f(k)$ for any polynomial $f \in \mathbb{Z}(x)$. In other words, given random $(\mathbb{G}_1, \mathbb{G}_2, e)$ generated by $\mathcal{IG}$, no efficient algorithm can solve BDH problem in $(\mathbb{G}_1, \mathbb{G}_2, e)$ with non-negligible advantage.

3.2. System Description and Authorization

Assume there is a Web services provider who provides a set of Web services and there are a number of users who have mobile agent enabled devices. Users can subscribe to or are assigned to any of the Web services and form several groups. Each group consists of the users who subscribe to a specific service resource. Assume the Web service provider (WSP) can act as a Key Distribution Centre (KDC) and have the secure channels to distribute keys to the users. Let l denote the cardinality of the Web service resources denoted as $r_1, r_2, \cdots, r_l$. All the users who subscribe to or are assigned to the same Web service resource form a group (G). Groups are denoted as $G[1], G[2], \cdots, G[l]$. For example, suppose that WSP provides several related Web services resource i.e. (r_1, r_2 and r_3). The corresponding mobile agent enabled user groups are $G[1]$ for accessing r_1; $G[2]$ for accessing r_2, and $G[3]$ for accessing r_3. Each user can subscribe to or be assigned one to all of the Web services resources. Based on our protocol, each client may join, leave a group or switch from one group to another dynamically. The integration of mobile agent and Web services or Mobile Web service can have three types of composition: parallel, sequential, and hybrid. Figure 2 shows a sequential composition, where as the request of a client, a mobile agent is dispatched to visit a set of WS providers where agent platforms are enabled.

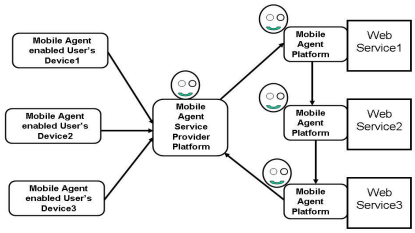

Figure 2. Mobile Agent-based Web Service Sequential Composition

3.2.1. System Setup The Web services provider (WSP) needs to set up the system such that all the necessary parameters can be used during the Web services lifetime. WSP selects the following parameters:

- a large prime $p = 2q + 1$ where q is also a prime;
- an additive group $\mathbb{G}_1$ and a multiplicative group $\mathbb{G}_2$ (both have order p);
- a master secret key $s \in \mathbb{Z}_q^*$; and
- a number $P \in \mathbb{G}_1$.

Based on the ID-based encryption algorithm [1], the WSP computes the system public key $P_{pub} = sP$ which is then sent to all members who have registered with WSP. WSP also selects two strong public one-way functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^*$.

Any user who wants to subscribe to or is assigned to any Web service has to register with the WSP and become a member first. We assume that there is a secure channel between each user and the WSP. The user applies to join the group and provides his/her ID. The WSP authorizes a privileged user by sending him/her a private key $s_{ID} = sQ_{ID}$, where $Q_{ID} = H_1(ID)$. After registration, users become members and they can subscribe to or are assigned to any Web services.

3.2.2. Subscription Assume members subscribe to or are assigned to some of the Web service resources $r_1, r_2, \cdots, r_l$. Consequently, the Web service provider can manage a $n * l$ matrix S as follows.

$$\mathcal{S} = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1k} & \cdots & S_{1l} \\ S_{21} & S_{22} & \cdots & S_{2k} & \cdots & S_{2l} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ S_{m1} & S_{m2} & \cdots & S_{mk} & \cdots & S_{ml} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ S_{n1} & S_{n2} & \cdots & S_{nk} & \cdots & S_{nl} \end{pmatrix} \qquad (3)$$

Where $S_{mk} = 1$ if the user u_m is a member of Web service $G[k]$, i.e. u_m is in group $G(k)$. $S_{mk} = 0$ if the user u_m is not a member of Web service $G[k]$, i.e. u_m is not in group

$G(k)$. n is the number of the users; l is the number of the services; k denotes the Web service $(1 \leq k \leq l)$; and m denotes the user u_m $(1 \leq m \leq n)$.

3.3. Mobile Agent and Web Services Authentication Scheme

In this section, we present an authentication scheme for the integrated system of mobile agent and Web services. As we mentioned in previous section, each user has a unique identification ID_i. No matter which user launched the mobile agent to the Web service system, the WSP can verify whether the mobile agent is really from the legitimate user or not.

3.3.1. Signature Scheme To sign a message $m \in \{0,1\}^*$, user U_i selects a random number $r \in \mathbb{Z}_q$, a generator $P \in \mathbb{G}_1$, and a public hash function $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and computes $R_i \leftarrow rQ_i$ and

$$S_i \leftarrow (H_2(m, R_i) + r)s_{ID_i} \tag{4}$$

The signature is now a triple (R_i, S_i, m).

To verify the signature, the following is checked:

$$e(S_i, P) \stackrel{?}{=} e(H_2(m, R_i)H_1(ID_i) + R_i, P_{pub}) \tag{5}$$

3.3.2. Authentication Scheme The scheme works as follows. The WSP sets up the system by following the algorithm presented in section 3.2.1 and then distributes each user's secret key via secure channels separately. At this stage, each legitimate user has a public key P_{pub}, secret signing key s_{ID_i}, and a unique identification ID_i.

Before launching the mobile agent, the user signs the message M_r using his/her secret signing key s_{ID_i} and generates a signature triple (R_i, S_i, M_r), where $R_i \leftarrow rQ_i$, $S_i \leftarrow (H_2(M_r, R_i) + r)s_{ID_i}$, and M_r is the message.

When the WSP receives the signed register message, it can verify the signature using the public key P_{pub} and the sender's ID_i.

$$e(S_i, P) \stackrel{?}{=} e(H_2(M_r, R_i)H_1(ID_i) + R_i, P_{pub}). \tag{6}$$

3.4. Mobile Agent and Web Service Security Scheme

In this section, we present the mobile agent and Web services security scheme. The WSP can encrypt the service data using the same encryption key for each service group. Only legitimate users can decrypt the encrypted session key or message. We present the encryption and decryption in following sections.

3.4.1. Encryption Let k denotes the k-th service and t_k denotes the total user number of group $G[k]$. Q_i denotes the user's (ID_i) public key and M_k can be a session key or message for $G[k]$. The WSP can compute the following parameters:

$$\begin{aligned}
&(Q_{v_{11}}, Q_{v_{12}}, \cdots, Q_{v_{1k}}, \cdots, Q_{v_{1l}}) \\
&= (Q_1, Q_2, \cdots, Q_n) \times \mathcal{S} \\
&= (Q_1, Q_2, \cdots, Q_n) \times \\
&\begin{pmatrix}
S_{11} & S_{12} & \cdots & S_{1k} & \cdots & S_{1l} \\
S_{21} & S_{22} & \cdots & S_{2k} & \cdots & S_{2l} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_{m1} & S_{m2} & \cdots & S_{mk} & \cdots & S_{ml} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_{n1} & S_{n2} & \cdots & S_{nk} & \cdots & S_{nl}
\end{pmatrix}
\end{aligned} \tag{7}$$

Therefore:

$$Q_{v_{1k}} = \sum_{ik=1}^{t_k} Q_{ik} \tag{8}$$

WSP sets up l $(t_k - 1) \times t_k$ matrices, which are defined as follows:

$$\begin{pmatrix} a_{2k} \\ a_{3k} \\ \vdots \\ a_{tk} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{9}$$

The WSP also generates $t_k - 1$ auxiliary keys for each group $G[k]$

$$Q_{v_{ik}} = (Q_{1k}, Q_{2k}, \cdots, Q_{tk}) \times a_{ik}, (2 \leq i \leq t_k) \tag{10}$$

which means:

$$\begin{aligned}
Q_{v_{2k}} &= Q_{1k} + Q_{2k} \\
Q_{v_{3k}} &= Q_{1k} + Q_{3k} \\
&\cdots \\
Q_{v_{tk}} &= Q_{1k} + Q_{tk}
\end{aligned}$$

For some random $r_k \in \mathbb{Z}_q^*$, the WSP can compute the following parameters.

$$U_{1k} = r_k P \tag{11}$$

$$U_{ik} = rQ_{v_{ik}}, (2 \leq ik \leq t_k) \tag{12}$$

$$V_k = M_k \oplus H_2(e(P_{pub}, r_k Q_{v_{1k}})) \tag{13}$$

The WSP sends the ciphertext $(U_{ik}, (1 \leq ik \leq t_k), V_k)$ to the set of users $\mathcal{U}_k$.

3.4.2. Decryption Assume that the user (ID_i) subscripts to kth Web service. The user can set a vector $a_{1k} = (0, \cdots, 0, 1, 0, \cdots, 0)$ with t_k elements where only the ith element is 1, Then $\mathcal{A}$ is a matrix.

$$\begin{pmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{tk} \end{pmatrix} \tag{14}$$

The user (ID_i) can solve the following system of equations.

$$(x_{1k}, x_{2k}, \cdots, x_{tk}) \times \mathcal{A} = (1, 1, \cdots, 1) \tag{15}$$

With $(x_{1k}, x_{2k}, \cdots, x_{tk})$, we can get

$$(x_{1k}, x_{2k}, \cdots, x_{tk}) \times \begin{pmatrix} Q_{ik} \\ Q_{v_{2k}} \\ \vdots \\ Q_{v_{tk}} \end{pmatrix} = Q_{v_{ik}} \tag{16}$$

To decrypt the ciphertext, the user (ID_i) needs to compute $e(P_{pub}, r_k Q_{v_{1k}})$. With knowledge of the private key s_{ID_i}, user can do via:

$$\begin{aligned} & e(P_{pub}, r_k Q_{v_{1k}}) \\ & = e(P_{pub}, r(x_{1k}Q_{ik} + x_{2k}Q_{v_{2k}} + \cdots + x_{tk}Q_{v_{tk}})) \\ & = e(P_{pub}, r_k x_{1k} Q_{ik}) \cdot e(P_{pub}, r_k(x_{2k}Q_{v_{2k}} + \cdots + x_{tk}Q_{v_{tk}})) \\ & = e(r_k P, x_{1k} s Q_{ik}) \cdot e(P_{pub}, x_{2k} r_k Q_{v_{2k}} + \cdots + x_{tk} r_k Q_{v_{tk}})) \\ & = e(U_{1k}, x_{1k} s_{ID_i}) \cdot e(P_{pub}, x_{2k}U_{2k} + \cdots + x_{tk}U_{tk}) \end{aligned} \tag{17}$$

Then, the user(ID_i) can compute

$$M_k = V_k \oplus H_2(e(U_{1k}, x_{1k} s_{ID_i}) \cdot e(P_{pub}, \sum_{ik=2}^{t_k} x_{ik} U_{ik})) \tag{18}$$

3.5. Re-keying for Member Changes (Members Join, Leave and Switch Web Service Groups)

In order to maintain the forward secrecy and backward secrecy, when a member changes to another Web services group $G[k]$, the Web service provider needs to re-key the corresponding Web service group session key. This includes cases where new members join, old members leave, and an existing member switches from one or several Web service groups to another one or several Web service groups.

Here we consider the case where one member switches from one Web service group to another. The join and leave operations are similar to the group switch operation. Suppose that a member u_m unsubscribes from the web service group $G[k]$ and subscribes to Web service group $G[k'](k \neq k')$, then the Web service provider needs to update the Web service group registration matrix S as shown below.

$$\mathcal{S} = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1k} & \cdots & S_{1k'} & \cdots & S_{1l} \\ S_{21} & S_{22} & \cdots & S_{2k} & \cdots & S_{2k'} & \cdots & S_{2l} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ S_{m1} & S_{m2} & \cdots & S_{mk} & \cdots & S_{mk'} & \cdots & S_{ml} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ S_{n1} & S_{n2} & \cdots & S_{nk} & \cdots & S_{nk'} & \cdots & S_{nl} \end{pmatrix}$$

Here S_{mk} is set to 0, and $S_{mk'}$ is set to 1.

Then the Web service provider needs to recompute $(Q_{v_{ik}}, ik = 1, \cdots, t_k)$ to revoke the member u_m from Web service group $G[k]$, and then recompute $(Q_{v_{ik'}}, ik' = 1, \cdots, t_{k'})$ to add the member u_m to data group $G[k']$. Then Web service provider can encrypt those two Web service session keys by computing $(U_{ik}, (1 \leq ik \leq t_k), V_k)$ and $(U_{ik'}, (1 \leq ik' \leq t_{k'}), V_{k'})$ correspondingly.

$$\begin{gathered} U_{1k} = r_k P \\ U_{ik} = r Q_{v_{ik}}, (2 \leq ik \leq t_k) \\ V_k = M_k \oplus H_2(e(P_{pub}, r_k Q_{v_{1k}})) \\ U_{1k'} = r_{k'} P \\ U_{ik'} = r Q_{v_{ik'}}, (2 \leq ik' \leq t_{k'}) \\ V_{k'} = M_{k'} \oplus H_2(e(P_{pub}, r_{k'} Q_{v_{1k'}})) \end{gathered}$$

The WSP sends the ciphertext $(U_{ik}, (1 \leq ik \leq t_k), V_k)$ and $(U_{ik'}, (1 \leq ik' \leq t_{k'}), V_{k'})$ to the set of users $\mathcal{U}_k$ and $\mathcal{U}_{k'}$.

The corresponding Web service group members need to decrypt the session key by using their ID-based private keys.

When a new member joins the group, the Web service provider only needs a new row in the registration matrix, then recomputes the related parameters which is similar to the member's switch into the new Web service group.

When an existing member leaves the group, the Web service provider needs to remove one line in the registration matrix or just set all values to 0, and recompute the related parameters which is similar to the member's switch away from one Web service group.

In some cases, we can keep some member IDs and all parameters. Thus, when the new members join, Web service provider will not need to perform the re-computation. For example, when new students enrolling at a University, the University can have students' ID numbers before the students come in.

3.6. Re-keying for Web Service (Web Services Provider adds or removes Web service)

When a Web service provider wants to add a new Web service, s/he only needs to add a new column in the registration matrix, then recompute the related parameters.

When the Web service provider wants to remove a Web service, the process is much easier; s/he only removes the corresponding column in the registration matrix.

4. Secure Mobile Agent and Web Service Applications

The integration of Mobile agent and Web services can be applied to many applications. It can be widely used in E-commerce, E-Government, military, hospital and remote education systems etc. Here we present a simple example to illustrate this security architecture.

Suppose that a university provides Web services for each subject. For the sake of simplicity, we assume that there are 5 subject resources in this semester: r_1, r_2, r_3, r_4, r_5. We also assume that there are 6 student users $(u_1, u_2, \cdots, u_6)$ and each student user has a mobile agent enabled device. Student user u_1 registers for 5 subject Web service resources r_1, r_2, r_3, r_4, r_5 ; student user u_2 registers for 3 subject Web service resources r_1, r_2, r_3, user u_3 registers for 3 subject Web service resources r_2, r_4, r_5; student user u_4 registers for 3 subject Web service resources r_1, r_3, r_5; student user u_5 registers for 2 subject Web service resources r_4, r_5; student user u_6 registers for 3 subject Web service resources r_2, r_4, r_5 (see table 1). Thus, the university has 5 Web service resource groups, group $G[1]$ has 3 student users u_1, u_2, u_4; group $G[2]$ has 4 student users u_1, u_2, u_3, u_6; group $G[3]$ has 3 student users u_1, u_2, u_4; group $G[4]$ has 4 student users u_1, u_3, u_5, u_6; group $G[5]$ has 5 student users u_1, u_3, u_4, u_5, u_6. Students can send their mobile agents sequentially to access the subject Web services they are registered or subscribed.

Student Users	r_1	r_2	r_3	r_4	r_5
u_1	1	1	1	1	1
u_2	1	1	1	0	0
u_3	0	1	0	1	1
u_4	1	0	1	0	1
u_5	0	0	0	1	1
u_6	0	1	0	1	1

Table 1. Example

In the Setup phase, the university Web service provider can set up the necessary parameters $p, \mathbb{G}_1, \mathbb{G}_2, P$ and the system master secret key s.

In the registration phase, all the student users need to register with the university Web services provider. Suppose that users are assigned their IDs with email $(u001, u002, \cdots, u006$@myuniversity.edu) (student user may use any email or other identity). Then the university WSP generates for each student user a private key (s_i) based on their identification (here is the email) and send to each student user separately .

The university Web service provider can manage a $5 * 6$ matrix as follows based on the student users and their registered or subscribed Web services.

$$\mathcal{S} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

At this stage, the university Web service provider can generate the encryption keys for each subject Web services resource based on the students registered or subscribed. When subject Web service message is encrypted with this encryption key, only student users who registered to this subject Web service can decrypt it.

For instance, student user u_4 is subscribed to 3 subject Web services resources r_1, r_3, and r_5. This user launches his/her mobile agent to the university Web service provider with signed message. This mobile agent uses email ($u004$@myuniversity.edu) as his/her identification, and sequentially moves to subject Web service resource platforms of r_1, r_3, r_5. In the first subject Web service resource platform, agent interacts with the host to get the subject Web service, the host verifies the identification using the authentication protocol, and then provides the service such as lecture notes, assignment etc. The Web services host encrypts the result with session key and transfer it to mobile agent. At same time, the host encrypts the session key with the subject Web service encryption key and sign it, and send it to the user's email. When the mobile agent moves to second and third subject Web service platforms, the same process will repeat.

When the student user receives his/her mobile agent, s/he gets the encrypted Web service data from three resources. Then, s/he receives the encrypted session keys from the email and decrypts these session keys with his private key s_4, and then use it to decrypt the encrypted Web service data. At same time, s/he also can verify that it is from the service provider by using the authentication protocol.

If a certain student user (e.g., u_5) wants to drop one subject and leave the subject Web service group G_4, the university Web service provider only needs to re-setup the Web service resource r_4 encryption key, and all other student users are not affected with their private keys unchanged.

If a new student user (e.g. u_7) joins in the subject Web service group (G_2, G_4), the university Web service provider can assign an identity (e.g. $u007$@myuniversity.edu) and a private secret key (i.e. s_7) to this student user, and then regenerate the subject Web service resource (r_2, r_4) encryption key. All existing student users are not affected.

If the university Web service provider wants to remove the subject Web service resource (r_1), they only need to send the canceled service message to the group student users (u_1, u_2, u_4).

If the university Web service provider wants to add one subject Web service resource (r_6), they need to setup a new encryption key based on the identification of users subscribed to subject 6.

To summarize, this scheme can provide a security solution for mobile agent Web service. As mobile agent could not take the secret key, the authentication can not be done with username/password pair. With this scheme, the mobile agent can use the owner's email as identification. The Web service provider can encrypt the Web services data with a session key, and encrypt the session key with corresponding Web service key, then email it to the user. Only legitimate users can get the session key by decryption. Then s/he can use the session key to decrypt the encrypted Web service data. They can use the authentication protocol to verify each other.

In addition, this scheme is scalable both from the server's side as well as from the requester's side; the Web service provider can add or remove Web service, and members can join or be removed. This scheme is stateless and any member leaving or joining the group will not affect other member's keys. This scheme is inherently traitor tracing because their members' keys are related to their ID. This scheme is flexible and efficient, as the members do not need much computational power because most computations are performed by the server.

5. Conclusion

In this paper, we have proposed a new mobile agent and Web services security architecture. In this security architecture, Web services security is based on an ID-based public key management algorithm. It provides authorization, authentication and confidentiality security service. With this scheme, mobile agent authentication can be done without using the username/password pair. This scheme provides an alternative to the current scheme without using Certification Authorities (CA) based-public key infrastructure. It can simplify the key management and reduce the computation particularly for group-oriented web services.

References

[1] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *Advances in Cryptology-Crypto 2001, LNCS*, 2139:213–229, 2001. Springer-Verlag.

[2] L. Chen, K. Harrision, N. P. Smart, and D. Soldera. Applications of multiple trust authorities in pairing based cryptosystems. *IEEE Transactions on Broascasting*, pages 260–275, 2002.

[3] D. Cooney and P. Roe. Mobile agents make for flexible web services. In *Proceedings of The Ninth Australian World Wide Web Conference*. Quensland, Australian, July, 2003.

[4] L. Danny B and M. Oshima. *Programming and deveioying Java Mobile Agents with Aglets*. Addison-Wesley, Massachusetts 01867, 2003.

[5] X. Du, Y. Wang, J. Ge, and Y. Wang. An identity-based encryption scheme for key distribution. *IEEE Transactions on Broascasting*, June:264–266, 2005.

[6] N. Gibbins, S. Harris, and N. Shadbolt. Agent-based semantic web services. In *WWW2003*, pages 710–717. ACM 1-58113-680-3/03/0005, May 2003.

[7] D. Greenwood and M. Calisti. Engineeringweb service - agent integration. In *IEEE International Conference on Systems, Man and Cybernetics (SMC 2004)*. The Hague, The Netherlands, October 2004.

[8] F. Ishikawa, N. Yoshioka, Y. Tahara, and S. Honiden. Toward synthesis of web services and mobile agents. In *Proceedings of the AAMAS2004 Workshop on Web Services and Agent-based Engineering (WSABE)*. New York, USA, July, 2004.

[9] C. Kaufman, R. Perlman, and M. Speciner. *Network Security Private Communication in a Public World*. Prentice hall PTR, Upper Saddle River, NJ 07458, 2002.

[10] R. Liu, F. Chen, H. Yang, W. C. Chu, and Y.-B. Lai. Agent-based web services evolution for pervasive computing. In *Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC04)*, 2004.

[11] Z. Maamar, F. Akhter, and M. Lahkim. An agent-based approach to specify a web service-oriented environment. In *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE03)*, 2003.

[12] E. Newcomer. *Understanding Web Services XML, WSDL, SOAP, and UDDI*. Addison-Wesley, 201 W. 103rd Street Indianapolis, IN 46290, 2002.

[13] M. O'Neil, P. Hallam-Baker, S. M. Cann, E. Simon, P. A. Watters, and A. Whiter. *Web Services Security*. McGraw-Hill/Osborne, 2600 Tenth Street Berkeley, California 94710, 2003.

[14] A. Padovitz, S. Krishnaswamy, and S. W. Loke. Towards efficient selection of web services. In *Proceedings of Workshop on Web Services and Agent-based Engineering AAMAS'2003*, 2003.

[15] R. Pascotto, B. Schiemann, and E. Kovacs. Giving mobile users access to net-based services a mobile agent approach. In *Flexsible Working*, pages 164–172. IOS Press The Netherlands, 2004.

[16] J. Peters. Integration of mobile agents and web services. In *The First European Young Researchers Workshop on Service Oriented Computing (YR-SOC 2005)*, April 2005.

[17] Y. Wang. Dispatching multiple mobile agents in parallel for visiting e-shops. In *3rd International Conference on Mobile Data Management (MDM2002)*, pages 53–60. IEEE Computer Society Press, Jan. 8-11 2002, Singapore.

[18] Y. Wang and J. Ren. Building internet marketplaces on the basis of mobile agents for parallel processing. In *3rd International Conference on Mobile Data Management (MDM2002)*, pages 61–68. IEEE Computer Society Press, Jan. 8-11 2002, Singapore.

[19] W. Zahreddine and Q. H. Mahmoud. An agent-based approach to composite mobile web services. In *Proceeding of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*. IEEE, 2005.