A Secure Agent-based Single Sign-On Scheme Supporting Web Services Home Network Environments

Woei-Jiunn Tsaur and Jia-Xin Wu
Department of Information Management, Da-Yeh University, Taiwan

Abstract - The number of services in home network environments has been growing increasingly, and therefore users must manage multiple user names and passwords daily. The previous works like SAML (Security Assertion Markup Language) standard and the commercial software called .NET Passport provide web services’ single sign-on function; however, the SAML system not only increases the heavy loading of servers but also costs the Internet flow. Furthermore, it may cause the potential attacks like the replay and man-in-the-middle attacks. Also, in .NET Passport the privacy of users may suffer the risk of eavesdropping. Therefore, this paper develops an agent-based single sign-on system in web services based home network environments. The proposed single sign-on scheme can reduce the number of communications among users and servers, enhance the security of home network services, provide the privacy of users, and promote the efficiency of system in the web services based home network environments.

Keywords: Home network, user authentication, single sign-on, mobile agent, web services

1 Introduction

With the increasing growth of the number of services in home network environments including a wide variety of personal equipment, servers, services and networked devices in the home [1]-[5], user authentication is extremely crucial for consumers to access home servers, web and broadcasting resources in the integrated home network environments [1]. In a traditional password authentication scheme, users must manage multiple user names and passwords daily for accessing multiple servers, and therefore it cannot be efficiently used for solving the user authentication of multi-server based home network services. The objective of efficient user authentication in multi-server based home network environments emphasizes that any user can obtain service granted from multiple servers without repeating registration to each server.

More and more platforms can provide integrated services from multiple servers with the development of web services. While each platform has its own authentication method, this will limit the efficiency of web services. Except for using the trusted third party to authenticate users, a “single sign-on (SSO)” scheme can be considered as another method. SSO allows users to log in only one site to obtain all the services provided in multiple servers. In the previous work, Jeong et al. [1] proposed an SSO scheme in which a mobile user offers his/her credential information to the home network for obtaining user authentication and access to another domain such as web servers to get related information using this authentication, based on the SAML (Security Assertion Markup Language) standard. However, because the authentication procedure of SSO is complicated, it is easily attacked by hackers, and may therefore encounter many security threats. Although the security standard SAML established by the Organization for advancement of Structured Information Standards (OASIS) can provide single sign-on function, it was pointed out that there is still some worry on security when using SAML for SSO [6], [7].

Basically, the SSO technology developed by commercial software can be practical for common enterprises to provide user authentication services. However, their SSO services are mostly based on single authentication framework, which uses one authentication server to manage a variety of user data and also to complete operations of user authentication. In such a way, the performance of the authentication server must be affected. On the other hand, SAML, Liberty Alliance and web services belong to the framework of multiple authentication centers. Their existing common platforms use the token method, but the security of token is weak because cookies must be saved in clients [8]. Therefore, in this paper we develop an agent-based SSO to enhance the security of user authentication in home network service environments. A mobile agent is a software entity that can autonomously migrate from one node to another in order to perform operations on behalf of the users in the system. Through this concept, network traffic and communication latency can be reduced. Thus, the proposed agent-based SSO scheme can minimize the number of communications among users and servers, and promote the efficiency of system as well as less delay in home network service environments.

The rest of this paper is organized as follows. In Section 2, we review the previously proposed related technologies. Then we present the proposed agent-based SSO scheme in Section 3. In Section 4, we give security and efficiency analyses of the proposed SSO scheme. Section 5 simulates the proposed agent-based SSO system to certify the results. Finally, some concluding remarks are presented in Section 6.
2 Related work

In this section, we first present several existing SSO solutions, and then several SSO schemes for web services will be discussed.

2.1 Single sign-on solutions

Existing solutions for SSO can be divided into five categories [7], [9]-[11], including (1) broker-based SSO, (2) agent-based SSO, (3) token-based SSO, (4) agent and broker-based SSO, and (5) gateway-based SSO. We will introduce them in the following. There is a server that manages the warranty and users’ name in the broker-based SSO. Broker is used for electronic ID saving and reading for the next request [12]. Using of central database has decreased on management costs and provided one suitable and independent third party for authentication, like Kerberos. The advantage of this solution is the concentrated method that provides central database for users’ management. However, it requires upgrading its existing program to meet its need. Besides, all the application systems and services will be affected if the authentication server cannot calibrate with this concentrated management framework.

There is an agent program to authenticate users’ identities for various different application programs in the agent-based SSO [11], [13]. This agent program will need to design for different functions [14]. For example, it can use passwords or encrypted keys to remove the authentication operations of clients, and be installed in servers as the translator between the authentication system and users, such as SSH. Its advantage is easy for transplantation, but it needs to design a protocol to communicate with previous application programs. Because the authority of agent program is not easy to set up [15]-[17], it is inconvenient for the agent-based SSO to manage.

The token-based SSO is the most widely used password authentication system [18], such as the log-in authentication of FTP and mail server, which are called single-factor password authentication. This is easy and more available method for users, but it has some incipient faults in security; such as easy to guess, seldom change of password, using same password in different systems, etc. As well, it can cause great damage if a password is stolen accidentally.

When the agent-based SSO and the broker-based SSO are combined, the flexibility and the central management existing in the former one and latter one, respectively, are the advantages, and can be also economic in modifying the web application program from the contribution of agent-based SSO.

2.2 SSO schemes for web services

1. Kerberos

Kerberos protocol is based on the trusted third party [12]. Its framework is divided into servers and clients. It assists to authenticate users’ identities in open network environments. Server only needs to provide management and distribute Kerberos Ticket, while clients’ calls on server according to user warrant on Kerberos Tickets for certifying ID. The time for both synchronization and its security are equally important, otherwise, attackers can re-attack by changing host time. Moreover, the internal web services of enterprises are not suitable in opening services and the complex communication program between server and client related to Kerberos.

2. SESAME

The full name of SESAME is Secure European System for Application in Multi-vendor Environment [19]. It is a European organization security project, which is considered to be a European Kerberos. SESAME is based on the GSS-API (Generic Security Services Application Programming Interface), which provides SSO and security in a distributed environment. Although SESAME and Kerberos are both based on the same model, they don’t copy with each other, and SESAME is added some new characteristics. Moreover, SESAME is different from Kerberos in operating. In Kerberos, users are first authenticated by an authentication server and get tickets from the server, and then obtain authorization from a variety of servers; in SESAME, the server called privilege attribute server distributes privilege attribute certificates to users to obtain the authorization for service requirements.

3. SAML

SAML a standard developed by OASIS based on XML elements. It provides authorization and warranty of data exchange in the transactions of business. It can act as a trusted third party that provides identity verification for millions of service providers. Not only it can be used for ID verification, authorization and SSO between web services but also it can eliminate the complexity of one-to-one relationship on the networks effectively, which can be the basic platform for most ID verification. SAML defines many security validations including authentication assertion, attribute assertion and authorization assertion, which can keep the security for XML framework and achieve SSO e-business.

Although SAML is a standard for the present web services, it has a lot of security vulnerabilities. Because of mutual authentication is not required, the man-in-the-middle attack will happen. Even though the specification requires integrity and privacy that an attacker won’t see or change the content, the attacker can use the browser to connect with destination sites, so the HTTP attack will happen due to the lack of verification and tag notification during transmission.

4. Liberty Alliance

Founded in 2001, Liberty Alliance is composed by more than 170 manufacturers. Its aim is to develop one specification that allows users to enter personal information once and then use several application services. At the first phase, Liberty Alliance proposed Liberty Identity Federation Framework (ID-FF), which manages user identification in enterprises and applications based on the open SAML standard. Web Services Framework (ID-WSF) is developed at the second phase, the
the aim of which is to build up ID service framework of inter-
connection, ID attributes sharing, ID service description and
search etc. Nokia and Vodafone have already developed it for
B2B business. The main aim for Liberty Alliance is that all
related companies will have to use the same way to achieve
jointed-ID managements. But problems will happen when
enterprises connect to their partners and clients whom might
operate in different systems, and that will bring difficulties in
management.

5. WS Security (WSS)
OASIS established WSS technology committee and related
rules in July 2002. WSS simplifies the data sharing from
different programs by XML. It improves connection of secure
systems using XML standards and defines extended standards
of SOAP to ensure privacy of information transmission
between services. WS-Security can be used in different
products, such as XML firewall, web service management
software and web security access system.

Summarized above, commercial software and Kerberos are
to use a single authentication center, while SAML, Liberty
Alliance and WS-Security belong to multiple authentication
centers. They all have potential threats in security.

3 Proposed agent-based single sign-on
scheme

Agent programs have the ability of saving user identities
and passwords when users log in a web server at the first time.
Then, when users want to log in other application servers, the
agent programs will retrieve the saved user identities and
passwords to access these services. That is, users do not need
to enter them again. In the following, we will propose an SSO
scheme in which a user offers his/her credential information to
the home network for obtaining user authentication and access
to another domain to get related information using this
authentication, based on a mobile agent.

3.1 System framework

Home network has a special server called home server. The
home server manages home appliances and connects the home
network to the external systems. In the home network service
environments, the proposed SSO framework consists of three
main parts that are shown in Fig. 1. In the following, we will
introduce the details of the web site, mobile agent and user.

1. Web Sites
Web sites are the service sites providing all kinds of
services for users in this framework. These web sites have
their own independent manners of user authentication and
authorization management, and authentication information
must be recorded in the system service unit. The user
authentication and authorization in these web sites have the
following functions:

(1) Authenticating common users’ log-in operations and
managing access rights.
(2) Verifying all requests from reliable web sites.

![Fig. 1. The proposed SSO framework](image)

(3) Defining users’ authorization sent by reliable web sites.

2. Mobile Agent
Mobile agent can take the responsibility to transmit log-in
information and authentication information from unreliable
web environments instead of ones that can be relied on. It also
assists users to achieve the aim of SSO by requesting services
from other former servers with the authentication. The content
of agent includes code, status and attributes:

- **Code:** program that defines the actions of agent.
- **Status:** variation of agent, such as keeping it active when the
  agent moves to another host.
- **Attributes:** description of agent, including owners, resource
  requirement, authentication key, etc. The agent
cannot modify these attributes.

The operating process by an agent between the web server
and user browser is shown in Fig. 2. When the web server
needs to communicate with a user, it will download the
program of agent and change its status and then transmit it to
the user by SOA-RPC. After the user received the agent, he/
she can start and connect with the agent; when it is finished,
the agent will send back to the web server, and the operating
process is completed.
A user can transmit one agent to a server through POST and change its status by "Get" instruction. In HTTP, transmission of data (requirement/response) is through MIME protocol. For agent transmission, the way we use here is similar to MIME but with the description of agent (including description language and agent type).

3. User
This means common users who use this type of systematic service within service web sites under this framework.

3.2 System procedure
For the convenience of description, the notations are defined as follows:
B: browser for user
S: source site
D: destination site
bs: connection from a user’s browser to a source site
Cid: channel ID
S?: judging if S is a source site
bs_cid: a secure channel from a user to a source site
A→, B: data transmitted from A to B via C

The processing steps of this system are shown in Fig. 3 including the registration phase and SSO phase.

Summarized above, the SSO algorithm is shown below:

XML-based algorithm: authentication

Input: Sign-on request
Output: Sign-on response

1. Using IE browser to show sign-on interface for entering ID / Password.
2. User’s computer requires random value for entering servers.
3. User inputs ID and password.
4. Server counts r = BHash(Random), transmits r to user’s computer and sets Session("r") = r
5. User’s computer counts:
6. user_spw = BHash(BHash(user_id | user_pass | S) | r)
7. User’s computer transmits user’s ID, role password to server by agent.
8. User signs in to source site.
9. Server judges whether SSO is successful by the following steps:
10. config = XQuery(doc("config.xml")/policy)
11. user_spw2 = XQuery(config/user_list/user[@user_id=user_id]/SPW)
12. if(user_spw == BHash(user_spw2 | Session("r"))
13. return Accept
14. else
15. return Reject
16. endif
17. Server clears Session("r").
18. Server provides services.

4 Security and Efficiency Analyses
In this section, we will analyze the security and efficiency of the proposed SSO scheme.

4.1 Security analysis
In the following, we will analyze the proposed scheme can withstand possible security attacks, including the impersonation, replay, and man-in-the-middle attacks. First, several notations are defined as follows:
SA: system authority
S: source site
D: destination site
U: user
U→S: a communication from a user to a source site
U→D: a communication from a user to a destination site
IDU: user’s ID
mw: warrant for mobile agent, including IDU, mobile agent ID, user’s public key, valid time, routing list and list of server authorization
req: log-in request from user, including host serial number and timestamp

Fig. 2. Agent operations

Fig. 3. System operating procedure
1. Impersonation attack

When an attacker wants to impersonate a legal user to log in a host by intercepting the log-in request \([IDU, mw, req]\), he/she cannot forge \(mw'\) to pass the authentication of the host, because he/she has no ability of getting the secret key shared between the legal user and SA. Besides, this system uses the warrant and timestamp to certify the validity of agent key for preventing an attacker from impersonating a legal user.

2. Replay attack

For preventing the replay attack, this system uses a timestamp to avoid attackers to replay the stolen log-in request from a legal user. No matter \(U \rightarrow D\) or \(U \rightarrow S\), the host serial number and timestamp are contained in the attacker’s log-in information, and thus this can prevent the replay attack effectively.

3. Man-in-the-middle attack

In this system, the agent and authentication servers encrypt messages by the public key. Unless an intruder successfully attacks the authentication server and acquires its secret key, the attacker cannot log in a site successfully within a valid time interval. The mutual authentication between an agent and a host can prevent the man-in-the-middle attack more effectively.

4.2 Efficiency analysis

This system can reduce the web transmission between a user’s browser and the authentication server effectively by using the agent technology. The user just needs to enter an account and password at the first time and then he/she can visit other service sources without entering the password again. The agent uses SOAP-RPC to log in and achieve the aim of SSO by entering an account and password once and transmitting twice on the web. On the other hand, SAML has to re-verify the validity of cookie or profile by turnaround service, and this needs seven times for data transitions, three of which are to get turnaround services from the destination site, one of which is to visit the destination site, two of which is the confirmation to the SAML host, and one of which is to provide service at last. As well, the SSO scheme from commercial software first needs to get authentication warrant, and then acquires service tickets by using Kerberos, which needs four data exchange processes. According to the aforementioned analyses, the proposed SSO scheme is more efficient than others. Besides, by the service of agent, a variety of sources on the networks can be distributed to each service host instead of single authentication server to enhance the efficiency of services.

4.3 Discussion

Based on the above analyses, the comparisons among the proposed scheme, commercial software and SAML are shown in Table 1. Commercial software may damage users’ privacy for enquiring his/her information, while the proposed scheme will not cause such a problem due to setting agents. Besides, SAML uses a cookie as an authentication pattern, and the cookie is usually saved in a user’s browser, which can cause the replay attack or man-in-the-middle attack. The technologies of agent and encryption used in the proposed system can improve the SSO security.

Cookie is considered as the only warrant for user’s log-in in SAML. Such cookies are easily to be imitated or guessed. Malicious users will be treated as legal users as long as they get the legal cookies for access authorization. The critical problem for commercial software in using Kerberos authentication is that time synchronization needs to be ensured for checking the validity of timestamp. In the proposed scheme, even thought the data of agent has been decrypted, the access can still be rejected due to insufficient valid time length provided for entering the system.

Besides, the reuse and open platform abilities of technologies are the most concerning aspects in the system development of web services. The use of the agent technology in this paper can meet these development requirements.

<table>
<thead>
<tr>
<th>Function</th>
<th>Proposed SSO</th>
<th>Commercial software</th>
<th>SAML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy protection</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Cookie may reveal privacy)</td>
</tr>
<tr>
<td>Central authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Security</td>
<td>Agent, single password, data encrypted</td>
<td>Single password, data unencrypted</td>
<td>Single password; data encrypted</td>
</tr>
<tr>
<td>Reuse of technology</td>
<td>Complete</td>
<td>Partial</td>
<td>Complete</td>
</tr>
<tr>
<td>Open platforms</td>
<td>Yes</td>
<td>No, only for Windows</td>
<td>Yes</td>
</tr>
<tr>
<td>Capability of system integration</td>
<td>Good</td>
<td>Poor</td>
<td>Good</td>
</tr>
</tbody>
</table>

Table 1. Function comparisons among the proposed SSO, commercial software and SAML
5 System Simulation

The simulation includes two hosts, a web server and an application program server, each of which is equipped with 1024M RAM and installed Windows server operation systems and IIS server. The client uses Windows Internet Explorer. Java Servlet is used for HTTP request and response, and JAXP is used for supporting XML Schema for XML process. Java based SAAJ (SOAP with Attachment API for Java) is for SOAP message process, and JAXR is for UDDI communication (Java API for XML Registries). JAX-RPC (Java API for XML-based RPC) is used for SOAP communication protocol and WSDL file description when processing RPC. Besides, JADE is the development tool for agent program. All of these are listed in Table 2.

Table 2. System environment and development tools

<table>
<thead>
<tr>
<th>System Environment</th>
<th>Development Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing HTTP Request &amp;</td>
<td>Java Servlet</td>
</tr>
<tr>
<td>Response</td>
<td></td>
</tr>
<tr>
<td>Processing XML</td>
<td>JAXP supported XML Schema</td>
</tr>
<tr>
<td>Processing SOAP Messaging</td>
<td>SAAJ</td>
</tr>
<tr>
<td>Communication with UDDI</td>
<td>JAXR</td>
</tr>
<tr>
<td>Processing RPC</td>
<td>JAX-RPC</td>
</tr>
<tr>
<td>Agent Program</td>
<td>JADE</td>
</tr>
</tbody>
</table>

In the beginning, a user registers to the home network server by http://127.0.0.1/SSOWebHomeNetwork/register.jsp, as shown in Fig. 4. After registration, the user can enter the home page of the home network services management, as shown in Fig. 5. When the user wants to enjoy the service of shopping mall, he/she may choose the "Shopping Mall" item in Fig. 5. Because this is the first sign-on, the system will turn to the sign-on interface, as shown in Fig. 6. After signing on, Fig. 7 will first show the user ID "ssodemo" has successfully passed the SSO authentication, and then the user can obtain the service of the shopping mall. When the user needs to use another service, e.g. auction, he/she may click the "Auction site" item in Fig. 5 to directly go to the auction site to enjoy its service by employing ID "ssodemo", as shown in Fig. 8.
6 Conclusion

An agent-based SSO for web services based home network environments is proposed in this paper. Its advantages are listed in the following:

(1) Providing an environment of integrating an agent with web services:
Combining secure agent platforms with web services tightly, agents can take the information entered by users while keeping the security within an agent platform, and this information can be protected by the authentication servers which use the encryption scheme to improve the requirement of secrecy.

(2) Reducing communicational cost among users and hosts:
When users want to access from one web site to another by using an agent, the proposed scheme can achieve the reduction of time consuming in communication among users and hosts due to its characteristics of automation and mobility as well as decreasing the occurrences on hackers’ attacks.

(3) Implementation of SSO for multi-server in reliable home network domain:
In web services, each server has its own authorized authentication scheme, and it is easy to implement SSO by employing the mobility of agent.

Acknowledgement

This work was supported by the National Science Council of Republic of China under contract numbers NSC 101-2219-E-212-001 and NSC 101-2221-E-212-006-MY3.

References


