A General and Systematic Approach to Web Mashup Security

Shraddha Sarraf
M Tech Scholar
School of Computer Science and Information Technology,
Devi Ahilya Vishwavidyalaya, Indore, Madhya Pradesh, India

Pankaj Jagtap
Lecturer
School of Computer Science and Information Technology
Devi Ahilya Vishwavidyalaya, Indore, Madhya Pradesh, India

Abstract - There are several new and innovative ways for developing software applications on the web. One such innovative ways is "Web Mashup" which allows users to create and develop new web applications by combining data and services from other web applications and data sources. Web mashups are created using several technologies such as Asynchronous JavaScript and XML (Ajax), Rich Site Summary (RSS), Representational State Transfer (REST), and Extensible Mark-up Language (XML). The presence of numerous online available data sources and services make mashup creation faster, easier and richer in data and content. Besides these advantages, there arises wide array of security issues while integrating diverse data and services gathered from diverse sources into a completely new environment. The security issues include user authentication, data confidentiality, data integrity, access control, Cross Site Request Forgery (CSRF), Cross Site Scripting (XSS) and many more. Many research papers aimed at providing and presenting approaches that resolve these security issues but still there arises a need of some general and systematic approach to resolve these issues. This paper aims at providing a general and systematic approach to web mashup security.

Keywords - Confidentiality, Cryptography, Mashup Security, Privacy, SOP.

I. INTRODUCTION

The rapid growth of Web 2.0 has resulted into a number of new techniques and methodologies for the creation of web applications. One such technique is the mashing up of data and services from several independent sources to create a new service. The term 'mashup' first became popular and appeared in music industry where disc jockeys compose and produce a new song by combining or mixing lyrics and background music from various different songs. Likewise, a web mashup can be defined as a web application that is created by combining and integrating required data and services from several different available sources in order to satisfy user needs and tasks through displaying it to them in a completely different and new manner on their computer screens. The presence of numerous online available data and services make the creation of mashups faster and easier since mashup involves the use of already available data and services to create new web applications.

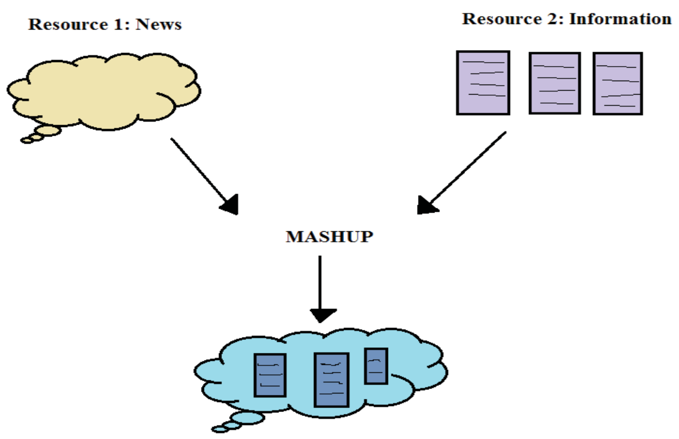

Figure 1: Architecture of a Web Mashup

One of the simple examples of the web mashups is that of a web page or a website that is extracting map information from one source and location information of a service such as hotels, shopping malls or houses for sale or rent from another source and displays the map with services provided by it as shown in Figure 1. Some of the popular web mashups are HousingMaps, Tracktor, SongsDNA and many more. HousingMaps is a mashup that applies real estate information, such as apartments for rent or homes for sale, from Craigslist to Google Maps. Similarly, Tracktor is a web mashup that helps you find pricing information that will help you in making purchasing decisions.

II. SECURITY OF WEB MASHUPS

Web mashups are primarily concerned with combining content and services from several different sources into a completely new web service, thereby introducing new security issues. These include: who made available the information to be combined, what are the intentions behind the availability of information, what is the trustworthiness of the content available? These security issues can arise either from the security problems or loop-holes of the technology being used in the creation of web mashups.

Web browsers implement a security model Same-Origin Policy which also creates security problems in web mashups. SOP utilizes origin to classify elements: elements belonging to the same origin can openly access each other's data and contents whereas elements from different origins are not allowed to access. Origins are identified by host name/Internet domain and protocol along with its port number. SOP imposes some curtailment on scripts and infringement of any of them may lead to security breach:

a) One origin script should not use XMLHttpRequest to interact with a website from some another origin.
b) One origin script should not be able to read or write data to/from another document or a frame belonging to some another origin.
c) Same-Origin Policy restricts scripts' access to cookies and web browser plug-ins.

Techniques used to create web mashup entirely refuse the SOP i.e. Same-Origin Policy. Hence, a new security framework is needed to be built where security issues like user authentication, data confidentiality, data integrity, access control are to be addressed.

a) User authentication:
User authentication is a process that allows a device to verify the identity of someone who connects to a network resource. In other words, it is the assurance that the communicating entity is the one that it claims to be.

b) Data confidentiality:

Data Confidentiality is whether the information stored on a system is protected against unintended or unauthorized access. Since systems are sometimes used to manage sensitive information. Or simply, the protection of data from unauthorized disclosure.

c) Data integrity:

Data integrity refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle. In other words, it is the assurance that data received are exactly as sent by an authorized entity i.e. contains no modification, insertion, deletion or replay.

d) Access control:

The prevention of unauthorized use of a resource i.e. this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do.

III. PROPOSED METHOD

The two most common attacks on web mashups are Cross-site scripting (XSS) and Cross-site request forgery (CSRF) are:

a. XSS is basically the result of session fixation which usually occurs when a new user's session is authenticated at the web server without invalidating the existing and previous session. This attack can be avoided by defining session timeouts and by regenerating session identifiers along with each request.

b. CSRF attack occurs when an intruder violates an authentic new user's session by invalidating the previous and existing session by capturing session credentials, redirecting and storing sensitive data and information. It can be avoided by enabling web servers so that they do not accept HTTP GET request, which is by far a practically impossible way. Thus, there arises a need of some other way to avoid CSRF attack and it is: with each HTTP GET and POST request, a token should be transmitted to the web server.

Most of the research work usually concentrates on providing solutions to XSS and CSRF attacks whereas the security issues like user authentication, access control, data confidentiality and data integrity have not been addressed properly. Thus, a general and systematic approach is needed to address and resolve the security flaws of web mashups. The proposed approach makes use of cryptographic techniques.

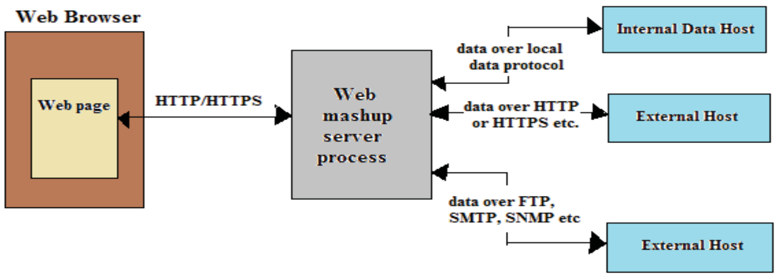
Web Browser
Web page
HTTP/HTTPS
Web mashup server process
data over local data protocol
Internal Data Host
data over HTTP or HTTPS etc.
External Host
data over FTP, SMTP, SNMP etc
External Host

Figure 2: Server-side mashup architecture

The proposed method focuses on server-side mashup architecture where data and services are combined on the server. This mashup architecture consists of two phases, namely, Phase 1 and Phase 2.

A. Phase 1:

Phase 1 comprises of three steps i.e. three-way handshake taking place between a user and a web mashup server as indicated in figure 3:

1. In step 1, a user transmits a message comprising of a request along with user ID, user's public key and a nonce

(a unique number generated with each session request). The transmitted message then has been encrypted using the public key of a web mashup server, which is known to the user.

2. In step 2, web mashup server responds with a message encrypted using user's public key. The message comprises of request-specific array of tokens for the various users, session timeout, session ID, a nonce (a unique number generated with each unique user request) and a function applied to the nonce sent by the user indicating that the user's request message is received. It is then properly and correctly interpreted by the mashup server.

3. In step 3, a user responds with a message encrypted using web mashup server's public key. The message comprises of a function applied to the nonce sent by the web mashup server indicating that the server's message is received and interpreted correctly by the user, along with this, there also exists request specific array of tokens sent by the web mashup server for the user indicating that the user has agreed on the server chosen request-specific array of tokens for various users.

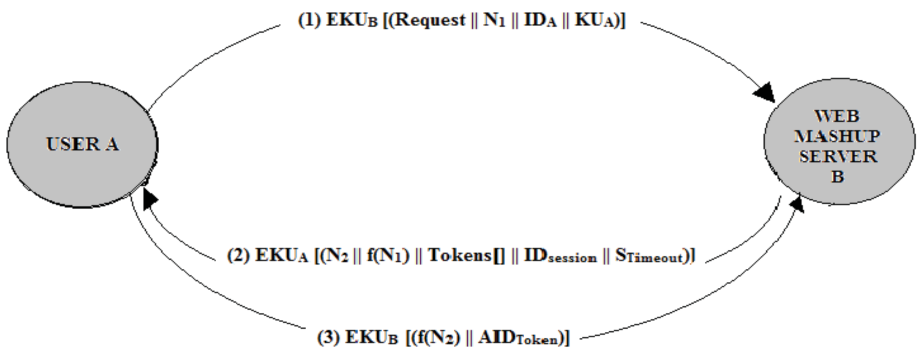
(1) EKU_B [(Request || N_1 || ID_A || KU_A)]
USER A
WEB MASHUP SERVER B
(2) EKU_A [(N_2 || $f(N_1)$ || Tokens[] || $ID_{session}$ || $S_{Timeout}$)]
(3) EKU_B [($f(N_2)$ || AID_{Token})]

Figure 3: Exchange of control information between a user and a web mashup server.

B. Phase 2:

Phase 2 comprises of two situations as indicated in Fig 4. Fig 4(a) indicates transmission of a request from a user to a web mashup server. Fig 4(b) indicates transmission of a web server's response to the user's request.

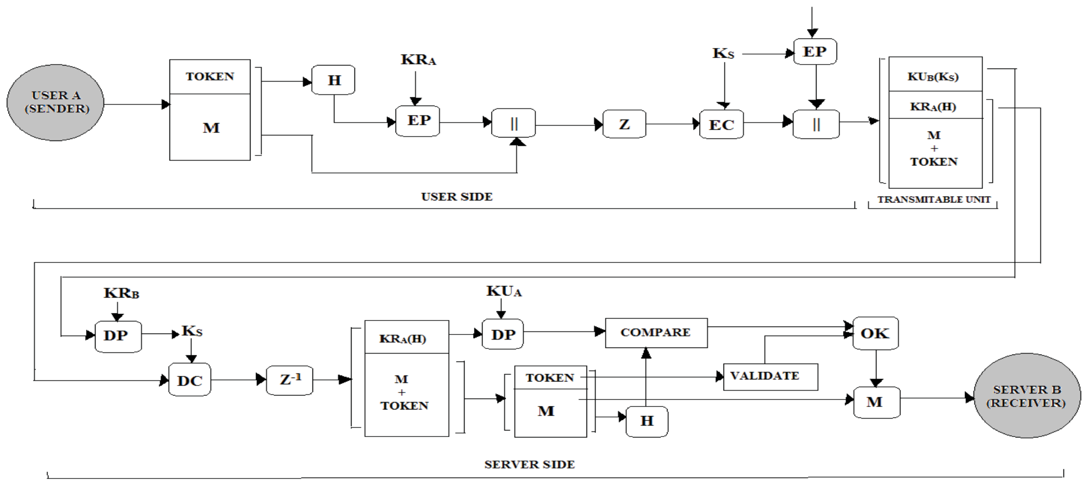

Figure 4(a): Transmission of a request from a user to a mashup server

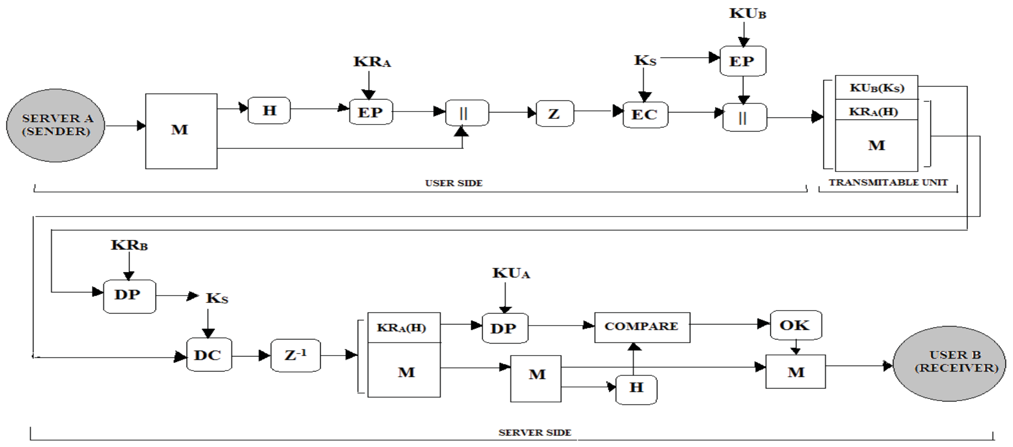

Figure 4(b): Transmission of a response from a mashup server to a user

1) User Authentication and Data Confidentiality

User authentication and data confidentiality is necessary for the protection of data and information to be exchanged between a user and a web mashup server. Both these security services are achieved in similar way in both the situations.

User Authentication - At sender's side, the calculated hash value is encrypted using the private key of a user i.e. sender (KR_A). At receiver's side, the encrypted hash value is decrypted using the public key of the user i.e. sender (KU_A).

Data Confidentiality - At sender's side, sender chooses a session key (K_S) and then encrypt the concatenated message using the chosen session key (K_S). The chosen session key (K_S) is also encrypted using the receiver's public key (KU_B), which is then combined with the encrypted concatenated message to be sent to the receiver. At receiver's side, first the encrypted session key (K_S) is extracted, which is then decrypted using receiver's private key (KR_B). The encrypted concatenated message is decrypted with the decrypted session key. Therefore, an unauthorized user having no knowledge of keys would not be able to disclose the contents of message, if he/she occupies the message.

2) Data Integrity

At sender's side, SHA-1 is used to generate hash code and the hash code is encrypted using the private key of the sender (KR_A). It is then concatenated with the message to be sent to the receiver. At receiver's side, the hash code is separated from the message; it is then decrypted using the public key of the sender (KU_A). The receiver calculates the hash code of the message received using the same SHA-1 algorithm. The received message will be considered as correct, accurate and accepted only if the generated hash code is same as the decrypted hash code; otherwise the received message will be rejected.

3) Access Control

This is achieved by configuring access control lists (ACLs) on the mashup server.

4) XSS Attack

XSS attack is concerned with invalidating the previous and existing session. To protect against the XSS attack, session timeout policy information sent by the mashup server in step 2 of phase 1 and the one accepted by the user in step 3 of phase 1 can be used.

5) Brute Force Attack

In Brute Force attack, in order to get access to a web page or a website, attacker tries all possible combinations of username and password. So, in the proposed method, if the attacker attempts brute force attack then his/her username will be blocked.

IV. CONCLUSION AND FUTURE WORK

With the introduction of Web 2.0, web application has entered in the new era of development. Mashups are suitable to build web applications and to develop new forms of visualizations with little knowledge of programming languages. There is also a feature of reusability in mashups which enables their faster creation instead of developing an application from scratch. Besides several advantages, there are some security issues like user authentication, data confidentiality, data integrity, access control too. The proposed method addresses these security issues efficiently. It also resolves XSS attack and brute force attack.

The proposed model, however, has the capacity to be extended. Security issues like non-repudiation, privacy are also necessary for the security of the system. These security issues are not addressed in the current proposed method and are left for the future work.

ACKNOWLEDGMENT

I am very grateful and would like to thank my guide Mr. Pankaj Jagtap and Prof. Dr. D.S Bhilare for their advice and continued support without which it would not have been possible for me to complete this work.

REFERENCES

[1] Ms.R.P Jadhav (2014, Nov) Security of Web Mashups: A Survey *International Journal of Innovative Technology & Adaptive Management* [Online].

[2] Koschmider, A; Torres, V; Pelechano, V. "Elucidating the Mashup Hype: Definitions, Challenges, Methodical Guide and Tools for Mashups." *International World Wide Web Conference*, Madraid, Spain, 2009.

[3] Murugesan, S. Understanding Web 2.0. *IT Prof.* 2007, 9, 34-41.

[4] Alexander Ritt, Philip Hortler "Security Aspects in Web 2.0 Mashup Systems" Austria.

[5] T. O'Reilly: "What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software", O'Reilly Media Inc., September 2005.

[6] Mashups: The What and Why- developer.force.com 2015.

[7] W.Wendy (2010, March), Web Mashup- A new breed of web application [online].

[8] W.Stallings, "Cryptography and Network Security Principles", Prentice Hall, 4th Edition, ISBN 0132023229, 2006

[9] Wikipedia, Mashup (Web Application Hybrid) [Online],Available:http://en.wikipedia.orglwikiMashup_(web_application_hybrid).