International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 38794-38797
© Research India Publications. http://www.ripublication.com

Digital Signed Mash up Security Framework

N. Angel,
Associate Professor, Department of MCA, St. Joseph's College of Engineering, Kanchipuram District. Chennai-600119
angel.mcastjosephs@gmail.com

R. Ilavarasan,
Senior Security Consultant, IGATE Global Solutions Pvt. Ltd, SIPCOT IT Park, Siruseri, Kanchipuram District. Chennai-603103
ilavarasan.rajendran@gmail.com

Dr. A. Chandrasekar,
Professor, Department of CSE St. Joseph's College of Engineering, Kanchipuram District. Chennai-600119
drachandrasekar@gmail.com

Abstract

A web mash up is a web application that integrates content from different providers to create a new service, not offered by the content providers. As mash up grows in popularity, the problem of securing information flow between mash up components becomes increasingly important. Web 2.0 has also introduced new possibilities for a better human computer interaction via rich internet applications such as mash up that provide a user-driven micro-integration of web-accessible data. In this paper, we present the policy-driven web mash up security tool that filters the original and unauthenticated packet by using the policy. It uses the mash up organization's privacy policies (MPP) along with digital Signature for the secure communication.

Keywords: web mash up, privacy policy, authentication, confidentiality

Introduction

With advancement in mash up source privacy policy it helps us to protect the data from the intruder. It plays a vital role and gives a backbone of the application security. Most of the application can be easily get traced and the private data can be easily stolen by the hackers. So many of the applications need to be protected the data from Man in Middle. The Man in Middle is nothing but the person who plays between the client and server in order to hack the system data while it passing through the network. They send the dummy or malicious packet along with the requested packet of the client from the server. If the intruder cannot understand the data, their dummy packets will provide the dummy web service. When the client uses to access the fake web service the hackers will easily steal the decrypted data from the client system.

Literature Survey

The Social Web is a set of social relations that link people through World Wide Web. This Social Web encompasses how the websites and software are designed and developed to support social relations. The new paradigms, tools and web services introduced by Social Web are widely accepted by internet users. The main drawbacks of these tools are it acts as independent data silos; hence interoperability among applications is a complex issue. This paper "Semantic web approach towards interoperability and privacy issues in social network" [1] focuses on this issue and how best it can use semantic web technologies to achieve interoperability among applications.

A popular Web programming paradigm, the mash up, has been widely adopted to quickly develop new Web applications by reusing existing third-party content. The word "mash up" also refers to a hybrid Web application that combines content from two or more sources to create a new service(s). Examples of mash up include third-party advertisements in web pages, Web widgets in portal sites such as iGoogle, and external code libraries. The paper "A Trustworthy Code Mash up Development Tool", handles the mash up security issues. The contributions of the paper are: (1) a description of JavaScript code mash up and its trust issues, and (2) a development tool (ToMaTo) for building trustworthy JavaScript code Mash up[3].

The Paper "Security of web mash up ", concrete requirements for building secure mash up, divided in four categories: separation, interaction, communication and advanced behavior control [6].

The concept of mash up, unfortunately, fundamentally contravenes the security model adopted by current web browsers, i.e., the Same Origin Policy (SOP). The policy prevents the documents or scripts loaded from one origin, defined as a combination of protocol, port and host, from accessing properties of a document from a different origin [8]. It is meant to protect web contents against cross domain attacks on the client side [9]. For a mash up, however, cross-domain communication among its components becomes a necessity. Without proper security controls in place, this opens the door to the attacks. To get out of this dilemma, both academia and industry are actively seeking effective solutions that permit but regulate the interactions among mutually-untrusting web services [7].

The mash up contains a component from bank, an advising component from a brokerage firm and an advertising component. The bank and brokerage component need to interact, to provide relevant advice regarding stock portfolio and interests; the brokerage and banking component provide the advertising component with keywords about financial habits, so that receive targeted advertisements. The bank component and brokerage component need to communicate with the servers of their firm, to retrieve the most recent information. The advertising component needs to communicate with servers from multiple advertising firms, to retrieve relevant advertisement.

Web 2.0 has introduced such applications to make relations among people. The examples are Facebook, twitter, Myspace, YouTube etc., One of the drawback of these social websites is that these tools acts as independent data silos. This paper has explains the issues relating to the social web and discusses about how can solve this with the help of semantic web technologies [1].

Problem and approach Overview

From end-users stand point the internal processes of Mash ups and their widgets are a black-box. Depending on the description of the Mash up and the trustworthiness of the Mash up creators, users may trust and reuse the existing Mash ups and widgets. This is due the basic principal of Mash up Architecture that facilitates the usage of services for non-expert users. However in the business domain, the trustworthiness of processes and tractability of results is important. For instance the output of a data-integration; Mash up should be originated from trustworthy resources or the Mash up should be created and / or verified by a certified Unit. As a result the Mash up components should be equipped with some trust indicators that fulfill the business requirements of different business processes and use cases of companies.

Mash ups may mislead employees into exposing too much data and unwillingly distributing sensitive company data. Especially, the easy access to data and services through such technology fosters data leakage. For users of such services and data, the trustworthiness proof of these sources is a challenging issue. If they do not know where the information comes from or who created the services, they cannot use this data for secure and meaningful decisions and statements. To define and discuss security and trust issues, the following facts have to be taken into account [13]:

- Mash ups are usually created by non-experts who are not aware of underlying data structure and security threats.
- Mash ups can be shared within the organizations and also with users outside.
- Mash ups are created by using data and information from known and unknown sources in-and outside the company.

Security Issues to be addressed

Techniques used to create web mash ups completely ignores the Same-Origin Policy and a web application violating the SOP would be a good working example of web mash up as an integrator would combine information from different content providers belonging to different trust domains. Hence a new security framework is required to be built where issues like data confidentiality, and user authentication are properly addressed:

1. User Authentication

The communicating entities may wish to guarantee and verify the identities of each other to build trust measures.

2. Data Confidentiality

The communicating entities may wish to guarantee that the content being exchanged should be readable to the intended parties only, any other third party on the way should not be able to disclose content of the packets being exchanged.

Proposed Solution

The proposed solution is designed as a tool which helps to improve security. This solution provides user authentication and data confidentiality by differentiating dummy and original packet. The proposed algorithm runs between client and server.

Architecture Diagram

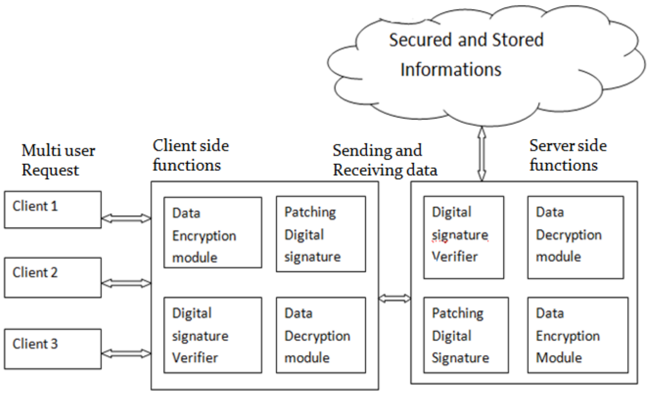


Fig.1. Proposed Architecture Diagram

The Fig 1 shows the System Architecture Diagram. The system architecture consists of three parts:

1) Secured information in cloud
2) Client request with patching digital signature
3) Server decrypter to respond the client

1) Secured information in cloud

a) In this, the original data from the client is initially read by the algorithm and they split in to several small packets with same size. This because to encrypt the data easily. Each packet is split into 128 bit size and they are allowed to encrypt in the next module. The size of the packet is initially converted in to kilo byte with that it separate each packet with the equal size. The equal partition is to make the easy encryption so that the performance of sending and receiving data will be more efficiently carried out in the algorithm.

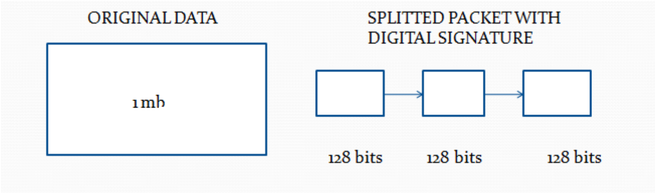
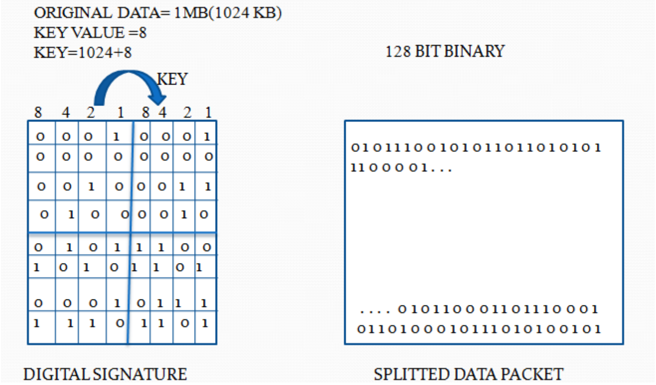


Fig 2. Initial Splitting of Data Packets

In Fig 2, the splitting of original packet is done by initially identifying the original packet size with that its partition into number of possible split packets. The split 128 bits packets are assigned with the digital signature. The patching of digital signature can be done in both front and back part of the split data. The digital signature is patched at the front side of the split data packet.

a) Creation of Digital Signature

The digital signature is act as an id to identify the original packed from the malicious packed. The digital signature is made with 8X8 metrics format which mention the size of the original data. The original data is converted in to binary format by using 8, 4, 2, 1 and assigned in 4 rows and columns. Then the key value 3 is added to the original size of data and then the size is also converted into the same binary format and assigned in next 4 rows and columns. Remaining 32 bits is assigned with top 16 bit and bottom 16 bits of split data packed.



8	4	2	1	8	4	2	1
0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	0	0
1	0	1	0	1	1	0	1
0	0	0	1	0	1	1	1
1	1	1	0	1	1	0	1

01011100101011011010101
1100001...

....010110001101110001
0110100010111010100101

Fig 3: Generating Digital Signature

b.Algorithm for generating digital Signature

STEP 1: Arrange the 128 bit in to two dimensional array order like{A00, A01, Ao3, ….A33}

STEP 2: SHIFT the ROW in sequence order in a way

```
Repeat I = 0 to I < = 9
Repeat j = i to j<=j+1
shift I th Row in I times
end Row change
end Row Selection
```

STEP 3: if i value is <=9 then

INVERSE the array column in a way

{A00, A10, A20, …A33}

2) *Patching Digital Signature*

When the packet from the client sent to the server the digital signature will attach with them when the server receive the packet it search for the digital signature whether it is authenticate one or not and respond to the client. When server sent any data to the client it will do the same process which has been done by the client [5]. The server also split the data packet in to small size of packets and each packet is encrypted and patched with the digital signature.

STEPS INVOLVED IN PATCHING DIGITAL SIGNATURE

STEP 1: Calculating using SIZE of ORIGINAL DATA

STEP 2: Create the Digital signature and Patch with each Data Packet

STEP 3: Sending the Data packet

3) Verifying Digital Signature

The main objective of module is to propose validation process which helps us to provide the security. This algorithm provides security and plays behind the application of client. This algorithm helps to differentiate the original and dummy packet in the multiple networks by adding digital signature in its each packet. When any data from the client or from the server the data will be patched with the digital signature so that the server or client able to understand the whether the packet is original or not.

This validation is done from both side client as well as server. It checks for the original size of the packet which is mentioned in the digital signature of the split packets. It checks the id which present in the digital signature with key value which present in both the client and the server. Its add the original size with the key value and verify whether the id which present in the digital signature is true or not and also it check the remaining 32 bits of data is belong to that split data packets by checking the first l6 bits and last 16 bits of the split packet. This verification is done on both the side client as well as server in order to identify that the data is belong to the original one or not.

When the intruder inject the dummy packet in the middle of multiple network in order to create a dummy web service in the client web browser or to send a dummy request to get the profile or any personal data of the person.

The intruder sent the packet along with original packet but the packet sent by the intruder will not satisfy the verification which takes place on both client and the server side. When the packet is considered as an unauthenticated then the server and client both does not respond to that packets. The client simply does not execute the packet and the server does not respond to the request of that packet.

STEPS INVOLVED IN VALIDATING DIGITAL SIGNATURE

STEP 1: Receive the packet from the server

STEP 2: Check for the size and key value with the digital signature.

STEP 3: Validate the relationship between data packet and the digital signature

STEP 4: Allow the packet with the authenticated data else drop the packet.

Performance Analysis

This algorithm works behind the screen, it will check the integrity of the packet received and user authentication. The performance analysis is the determination of the number of resources (such as time and storage) necessary to execute them. Most algorithms are designed to work with inputs of arbitrary length. Usually the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity).

Packet Delivery Ratio

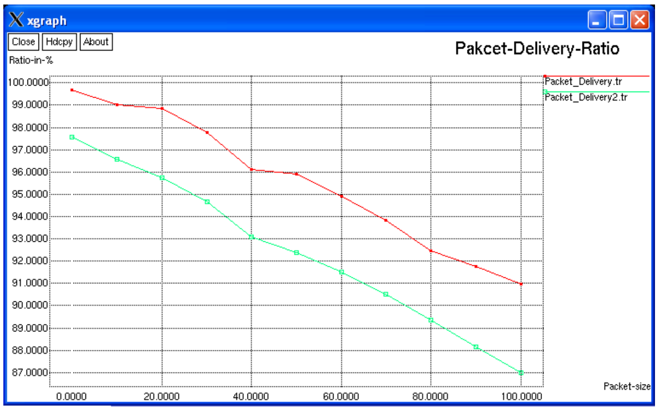


Fig. 4. Packet Delivery Ratio

In the chart given explains about the percentage of server load. When the amount of packet sent to the server is raised then the accuracy and speed of the packet delivery get reduced. The Y axis mentioned the percentage of accuracy of an packets sent whereas the X axis mention the size of the packet travelled across the network.

Time-Delay Ratio

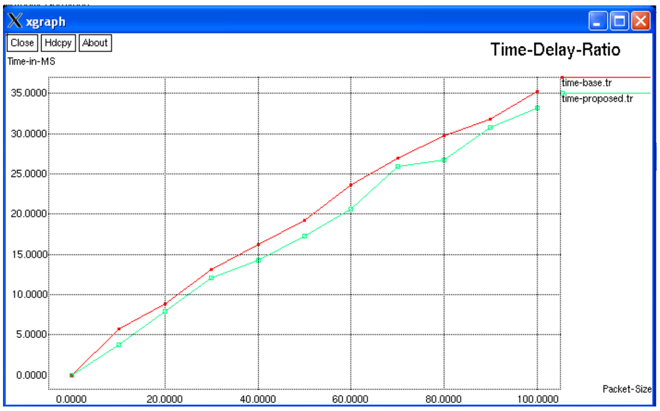


Fig. 5. Time Delay Ratio

The Fig. 5 explains the Time-Delay ratio of the packet which transmitted. The X axix represents the packet size and Y axis represents the time taken to deliver in milli seconds.

Conclusion

In this paper we introduced the framework for recharging the sensor nodes by sensors. This approach increases the longevity of the network. Simulation are performed to evaluate the performance of the proposed system in large scale network, and it shows the significant performance gains with respect to various metrics such as average response time, delay and network lifetime. The energy information of sensor nodes can be received by sensors in faster manner, and thus nodes are recharge soon. Thus the lifetime of the wireless sensor network has been prolonged.

References

[1] Dr V.KiranKumar, "Semantic Web approach towards Interoperability and privacy issue in social networks" International Journal on Web Service.", 2014

[2] Jian Chang, Krishna Venkatasubramanian, “A Trustworthy Code Mashup Development Tool”, University of Pennsylvania Scholarly Commons, 2011.

[3] Soon Ae Chun, Janice Warner and Angelos D. Keromytis, " Privacy Policy-driven Mashups, 2012

[4] Miguel Crespo; Stephen Brewster, “cyber technology policies to information systems”, International Journal of Business and Risk Management, 2013

[5] A. Berglund, S. Boag, D. Chamberlin, ” Xml path language (XPATH)” In InfoSecCD Conference., 2007

[6] Philippe De Ryck, Maarten Decat, Lieven Desmet, Frank Piessens, and Wouter Joosen, “Security of Web Mashups”, 2012

[7] Zhou Li, Kehuan Zhang, XiaoFeng Wang, “Mash-IF: Practical Information-Flow Control within Client-side Mashups”, IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), 2012

[8] J.Ruderman. The same origin policy. http://www.mozilla.org/projects/security/components /same-origin.html, 2008.

[9] C. Karlof, U. Shankar, J. D. Tygar, and D. Wagner. Dynamic pharming attacks and locked same-origin policies for web browsers. In Proceedings of the 15th ACM conference on Computer and communications security, pages 58–71 ACM New York, NY, USA, 2007.

[10] G. Radhamani, G.S.V. Radha Krishna Rao, Web Services Security and E-Business, ISBN: 978-1599041681, 2007

[11] Jackbe company website, http://blogs.jackbe.com/2008/03/mashupsecurity-101.html, last visited May 2010