ISSN: 2455 - 9172

EIJO Journal of Engineering, Technology And Innovative Research (EIJO–JETIR)

Einstein International Journal Organization (EIJO)

Available Online at: www.eijo.in

Volume – 1, Issue – 1, March – April 2016, Page No. : 49 - 53

An Efficient Identity Based Group Key Distribution Scheme for Client Side Security in Web Applications

J.Abinaya[1], R.Ranjitha[2], S.Swetha[3] and S. Manikandan[4]

[1, 2, 3] IV IT, Department of IT, EGSPEC, Nagapattinam, Tamil Nadu, India

[4] Assistant Professor & Head of IT, Department of IT, EGSPEC, Nagapattinam, Tamil Nadu, India

E-Mail: manikandan@egspec.org

ABSTRACT

The main purpose of this paper is to provide secure link and confidentiality of data within the group. In order to achieve this we primarily focus on the authentication protocols, which is the main region of attack by any hackers or intruders. Since we focus on authentication protocols we directly eliminate the chances of break through into the system. Here, we present WebIBC, which integrates public key cryptography into web applications without any browser plugins. The public key of WebIBC is provided by identity based cryptography, eliminating the need of public key and certificate online retrieval; the private key is supplied by the fragment identifier of the URL identifier. The implementation and performance evaluation demonstrate that WebIBC is secure and efficient both in theory and practice. WebIBC integrates identity based cryptosystem into web based applications and is totally established by JavaScript without any browser plugins.

Keywords: Public key Infrastructure, Public key Cryptography, Public Key Generator, Identity Based Cryptography, Limitation and Performance matching

1. INTRODUCTION

Public key cryptography is a fundamental building block for information security that can provide authentication, authorization, integrity and non-repudiation. But public key cryptography is seldom utilized in web applications. The mainstay of the project is to collaboratively generate a common key for peer to peer group communication. To dynamically perform re-keying operation after batch of joins or leaves using Queue Batch algorithm and to share resources using the generated group key. While acquiring ease of use services, users will have to give the control of their data privacy to the application providers [1]. Although application providers announce that these private data will not be abused and will be automatically handled without the involvement of administrators, these applications did not provide any mechanisms to guarantee this promise. Users have to trust the providers to be reliable and honest, and will "do no evil". But some providers have "done evil". One famous example is Yahoo providing user information in its email system to government that helped land a journalist in prison for 10 years [3]. And the leakage of private information will bring greater harm to enterprise users. Some providers like Google and Yahoo also provide services such as Google Apps for enterprise users to take the place of their own email servers and applications. The misuse of provider's privilege will bring huge losses for their customers.

With the increasing popularity of Web 2.0 applications like Google Gmail and Google Docs, people are moving their private data and communication information from their local storage to the online application providers. These online applications offer reliable storages and ease to access services. With the AJAX techniques these applications only rely on browsers with common features including HTML, JavaScript and CSS, without the need of installing any browser plugins or software. These applications make the exchange, management and access of data much simpler than previous desktop applications [2],[4].

Network services are provided by means of dedicated service gateways, through which traffic flows are directed. Existing work on service gateway placement has been primarily focused on minimizing the length of the routes through these gateways. Only limited attention has been paid to the effect these routes have on overall network performance. These networks consist of various components like routers and gateways. But routers are not reliable since it has major disadvantages like packet lose, delay in data transfer due to traffic in network [6],[7]. To overcome these problems, the service gateways are used instead of routers in the network. The service gateways work under two algorithms. This paper is organized as follows: in Section 2 we introduce WebIBC with some background information, followed by the description of the system architecture and implementation in Section 3, and then the performance and security evaluation in Section 4. At last we conclude the paper and introduce future work in Section 5.

2. Challenges

The first challenge is how to get the recipient's public key. In traditional PKI, a sender needs to visit an online

- database to find recipients' public keys and certificates.
- For the same reason, it is hard to import private key into JavaScript programs. For some solutions, a plugin developed with native language will create a bridge between the browser and the local system.

- JavaScript can only access contents inside the pages from the same origin, which mean JavaScript cannot access a LDAP from another server or access local public key database.
- The sender must keep a local database including all possible recipients' public keys, like PGP. But for web applications, none of these methods are practical.
- The plugins, for example, IE ActiveX, will provide a JavaScript object as the interface to access a local file or cryptography devices, such as smart card and USB secure token, which are applied in some e-bank systems.

3. Contribution

In WebIBC, two mechanisms are integrated to resolve the above challenges and provide security and privacy for client side web users. The first one is Identity Based Cryptography (IBC), a type of public key cryptography in which the public key can be an arbitrary string. WebIBC can provide public key encryption and digital signature for the web applications without the need of online searching and retrieving of public keys or certificates [5].

In order to overcome the problems in the existing system the various components and algorithms were used in the proposed system. Here instead of routers the service gateways have been used which is very much reliable for the transfer of data through network. Using the algorithms the best gateway saw well as the best paths were found that reflects in reliability and reduction in time delay for data transformation [9].

Because the recipient's email address, which also serves as his public key, can be easily read from the HTTP form in the message sending web page, the JavaScript implementation of IBC can make WebIBC easily integrated into any web applications, and run in all browsers even text based http clients with JavaScript extension. The other is to provide the private key from the URL fragment identifier. In WebIBC, the private key is encoded into the fragment identifier component of the web application URL.

4. Identity Based Cryptography

Identity-based cryptography (IBC) is a form of public key cryptography for which the public key can be an arbitrary string, including email address, domain name, phone number and user name. The concept was used to eliminate the complexity of public key and certificate management. In a scenario that mani wants to send a message to palani at *palani@domain.com*, Mani will not need to retrieve Palani's public key and certificates from an online LDAP (Lightweight Directory Access Protocol) server or from a secure channel, but she just simply encrypts the message with Palani's email address "*palani@domain.com*" by an identity based encryption (IBE) scheme. And Mani can decrypt the message with the same scheme. IBC can be classified into Identity Based Encryption (IBE), Identity Based Signature (IBS) and Identity Based Authenticated Key Agreement protocol, or classified by the complexity assumption. After the concept was first suggested, IBS schemes were sooner founded, but IBE scheme remained a more challenging problem. IBE (BF-IBE) based on pairing was suggested After that, a series of IBE schemes were proposed such as. In the web browser and JavaScript environment, the schemes based on pairing are too complex and over kill. These schemes require at least 512 bits elliptic curve while only provide security similar to 160 bits Elliptic Curve Diffie-Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA). This is the motivation for selecting Combined Public Key (CPK) cryptosystem as our IBC scheme. A revised version of CPK algorithm is introduced here, which simplifies the origin one but still remains the security.

5. Combined Public Key Algorithm

We consider two public key based distributed combined algorithms, or interval-based algorithms for short, for updating the group key:

- Elliptic Curve Diffie-Hellman (ECDH)
- Elliptic Curve Digital Signature Algorithm (ECDSA)

Performance of these two public key based distributed combined algorithms under different settings, such as different join and leave probabilities, is analyzed. We show that the interval-based algorithms significantly outperform the individual re-keying approach and that the Queue-batch algorithm performs the best among the two interval-based algorithms. More importantly, the Queue-batch algorithm can substantially reduce the computation and communication workload in a highly dynamic environment. We further enhance the interval-based algorithms in two aspects: authentication and implementation. Authentication focuses on the security improvement, while implementation realizes the interval-based algorithms in real network settings. Our work provides a fundamental understanding about establishing a group key via a distributed and collaborative approach for a dynamic peer group.

5.1 ELLIPIC CURVE DIFFIE – HELLMAN (ECDH)

To efficiently maintain the group key in a dynamic peer group with more than two members, we use the tree-based group Diffie–Hellman (TGDH) protocol. Each member maintains a set of keys, which are arranged in a hierarchical binary tree. We assign a node ID to every tree node. For a given node, we associate a secret (or private) key and a blinded (or public) key. All arithmetic operations are performed in a cyclic group of prime order with the generator.

5.2 ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM (ECDSA)

We develop two interval-based distributed re-keying algorithms (or interval-based algorithms for short), termed the Rebuild algorithm, the Batch algorithm, and the Queue-batch algorithm. Interval-based re-keying maintains the re-keying frequency regardless of the dynamics of join and leaves events, with a tradeoff of weakening both backward and forward confidentialities as a result of delaying the update of the group.

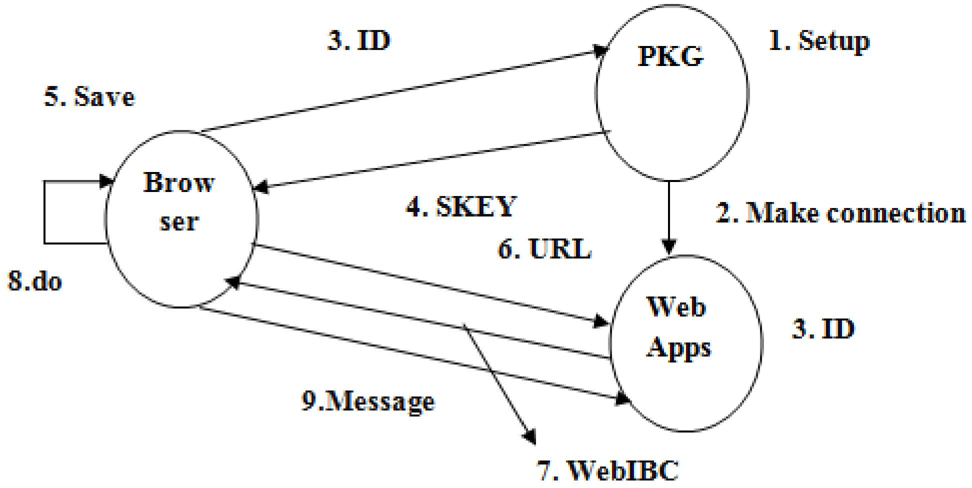


Figure 1.1: Working Flow Model.

6. Working Flow

In a scenario that Mani wants to send a secure message throw a web mail enhanced by WebIBC, the working flow is as follows:

1. The authority trusted by Mani and Palani establishes a PKG, which will generate the system parameters including the public matrix.
2. Web application embeds WebIBC into these systems together with the public system parameters released by the PKG.
3. Mani registers to the PKG with his ID.
4. PKG returns Mani’s private key. The communication between users and PKG requires a secure channel for authentication and the transmission of keys.
5. For example, PKG can encrypt the key by a password appointed by Mani and send it to the email address of Mani. Mani can append the private key as an fragment identifier to the Web application’s URL, then save it as a bookmark into the browser.
6. Now Mani can use this bookmark to log into the web application. It should be noted that the browser will send the URL without the fragment identifier, so the private key is secure.
7. The WebIBC JavaScript files will also be downloaded from the server, including the public matrix of system.
8. Mani uses this web application as normal, entering Palani’s email address and message content into the form. When Mani presses the send button, WebIBC JavaScript programs will get the email address from the form, and extract Palani’s public key from the matrix, then use this public key with Elliptic Curve Integrated Encryption Scheme (ECIES) to encrypt the message. If Palani wants, she can also generate an ECDSA signature with the private key imported from the bookmark URL fragment identifier component.
9. And then the message will be sent to the server.
10. Because the message has been protected, the Web application can do no evil to the message but only forward it to Palani. Palani can also login into his web application and decrypt the message by his private key in the fragment identifier and verify the message through the public matrix, similar to Mani.

7. Performance and Optimization

In this section, computation and bandwidth overheads of WebIBC are evaluated from tests and analysis at first. The benchmark result shows that the performance of our proof of concept implementation is acceptable for both users and browsers. Optimization methods for algorithm and programming practice introduced later can enhance the efficiency of WebIBC immensely; the estimated delay will be unnoticeable to users. The security of WebIBC under some possible attacks will also be discussed at the end of this section.

7.1 Possible Optimization

Reduce ECC key length from 192 bits to 160 bits, which provides the security similar to 1128 bits RSA. The amount of MUL operations will be reduced to 83% of 192 bits key. The time for every MUL will also be reduced.

Faster ECC algorithm. In our prove of concept implementation, the slowest point scalar multiply algorithm is used. In [8] some optimized algorithms are introduced and for encrypt and sign operations, 66% of the scalar multiply can be pre-computed.

7.2 Benchmark Result

The computation of WebIBC is mainly coming from three basic cryptography building blocks: AES, SHA-1 and big integer module Multiply (MUL) operation. The MUL is the main computation component of ECC and CPK. When the unit performance is known, the total running time can be estimated for specific factors. We run a benchmark include SHA-1, AES with 128 bits key and 192 bits big integer MUL on different browsers, so we can estimate the total computation overhead of WebIBC with different data sets and system parameters. The unit of SHA-1 and AES result is bytes per millisecond, the unit of MUL is operations per millisecond.

Comparison Results	SHA - 1	AES	CPK(MUL)
Internet Explorer	20.41	13.44	1.12
Opera	22.52	15.25	2.09
Netscape	32.43	20.55	3.54
Firefox	29.23	22.32	5.11

Table 1: Benchmark Results for Various Web Browsers

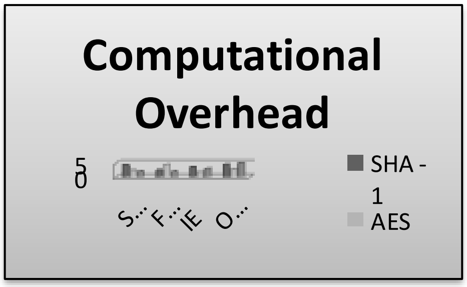
Techniques. Although the cryptography operations are very time consuming, this overhead can be neglected compared to user's editing performance, including thinking and writing the message. With Web 2.0 JavaScript techniques, these operations can be separately finished during the long period when user editing the message. The performance of AES and SHA-1 (about 20KB per seconds) is much better than people's editing speed.

8. Limitation

There are still some limitations in our system, including: the JavaScript is provided by the service provider, so it might be modified. And for an email application, the attachment is hard to be protected by WebIBC.

8.1 Phishing

Phishing attacks use both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. Social-engineering schemes use "spoofed" e-mails to lead consumers to counterfeit websites designed to trick recipients into divulging financial data such as credit card numbers, account usernames, passwords and social security numbers. Hijacking brand names of banks, e-retailers and credit card companies, phishes often convince recipients to respond.



8.2 Server Cheating

One possible threat is coming from the application server. In WebIBC, the cryptosystem implemented by JavaScript is embedded in the application pages that downloaded from application HTTP servers. The security of WebIBC depends on the reliability and correctness of the JavaScript program and data. If the server provides fake script, the security will be broken.

9. Conclusion and Future Works

In this work, The key agreement setting is performed in which there is no centralized key server to maintain or distribute the group key. We show that one can use the TGDH protocol to achieve such distributive and collaborative key agreement. To reduce the re-keying complexity, we propose to use an interval-based approach to carry out re-keying for multiple join and leave requests at the same time. We present WebIBC to protect the client side security and privacy of web applications. WebIBC integrates identity based cryptosystem into web based applications and is totally established by JavaScript without any browser plugins. We have implemented a prototype of WebIBC and performance evaluation indicates its effectiveness and efficiency over BF-IBE. The security analysis shows that WebIBC is resilient to some known attacks using the proposed schemes. The future work of WebIBC is to evaluate the feasibility of other IBC schemes on WebIBC, especially BF-IBE. In future fault tolerance can be implemented in the system. Further Efficient Authentication protocols can be implemented in the system. This software can be further enhanced with efficient protocols and cryptography techniques to enhance the operability and reliability of the system.

10.References

[1] B. Adida. Beamauth: two-factor web authentication with a bookmark. In CCS ’07: Proceedings of the 14th ACM conference on Computer and communications security.

[2] X. Boyen. General ad hoc encryption from exponent inversion be. In LNCS 4515, Springer- Verlag,pages 394–411,2007.

[3] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifier (URI): General syntex.2005.

[4] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. Proceedings of Crypto 2004, LNCS. Springer-Verlag, 2004.

[5] K. Paterson. ID-based signatures from pairings on elliptic curves, cryptology eprint archive, report 2002/004.

[6] A. Shamir. Identity-based cryptosystems and signature schemes. Crypto ’84, pages 47– 53, 1985.

[7] W. Tang, X. Nan, and Z. Chen. Combined public key cryptosystem. Proceedings of International Conference on Software, Telecommunications and Computer Networks (Soft- COM’04), 2004.

[8] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. Lecture Notes in Computer Science, 2139, 2001.

[9] C. Cocks. An identity based encryption scheme based on quadratic residues. Lecture Notes In Computer Science, 2260:360–363, 2001.

[10] B.Schneier. Applied cryptography: Protocols, algorithms, and source code in c, second edition. 1996.