IDENTITY BASED CRYPTOGRAPHY FOR CLIENT SIDE SECURITY IN WEB APPLICATIONS (WebIBC)

Archana B.Kadga

Assistant Professor at SMSMPITR, Akluj Dist, Solapur (Maharashtra), Karnataka, India

Abstract

The growing popularity of web applications in the last few years has led users to give the management of their data to online application providers, which will endanger the security and privacy of the users. In this paper, 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 inspired by BeamAuth [6]. The implementation and performance evaluation demonstrate that WebIBC is secure and efficient both in theory and practice.

Keywords: *PKC, Public key Infrastructure (PKI), Public Key Generator (PKG).*

----------***----------

1. INTRODUCTION

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 [21] 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.

While acquiring ease of use services, users will have to give the control of their data privacy to the application providers. 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 [1]. 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.

1.1 Related Work

Public key cryptography based solutions for the desktop counterpart of the above web applications have been deployed widely for many years. PGP [22] and S/MIME [18] are two de facto standards, and have been implemented within applications inside many desktop mail clients. The key management of these solutions require ad hoc trust management such as PGP "Web of Trust" or centralized Public Key Infrastructure (PKI). Generally, these methods can be classified into desktop software and browser plugins. A collection of these tools are listed in [2].

1.2 Challenges

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 challenges are twofold:

1. 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. Or 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. In web browsers, JavaScript programs are restricted in a sandbox. JavaScript can only access contents inside the pages from the same origin, which means JavaScript cannot access a LDAP from another server or access local public key database.
2. For the same reason, it is hard to import private keyinto JavaScript programs. For some solutions, a plugin developed with native language will create a bridge between the browser and the local system. The plugins, for example, IE ActiveX, will provide

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.

1.3 Our 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. 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, such as w3m [5] and lynx [3]. The other is to provide the private key from the URL fragment identifier, the substring starting from the first "#" symbol of a URL. In WebIBC, the private key is encoded into the fragment identifier component of the web application URL.

1.4 Paper Organization

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. WEB IBC BASIS

This In this section, we will first introduce IBC and fragment identifier in details, and then illustrate the system model of a WebIBC protected web application.

2.1 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 first introduced by Shamir in 1984 [19], used to eliminate the complexity of public key and certificate management. In a scenario that Alice wants to send a message to Bob at bob@domain.com, Alice will not need to retrieve Bob'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 bob's email address"bob@domain.com" by an identity based encryption (IBE) scheme. And Bob 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 based on [10]. After the conceptwas first suggested, IBS schemes [19, 17, 14] were sooner founded, but IBE scheme remained a more challenging problem. Until 2001, the Boneh-Franklin IBE (BF-IBE)based on Weil pairing was suggested [9]. After that, a series of IBE schemes were proposed such as [12, 8, 20].

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) [20] cryptosystem cryptosystem as our IBC scheme. A revised version of CPK algorithm is introduced here, which simplifies the origin one but still remains the security.

2.2 System Setup

In the setup procedure, a trusted authority will generate a master secret in the system and public parameters know to all entities. Every entity needs to authenticate him to authority, and the authority will extract the private key from the master secret according to entity's identity. In CPK, the authority providing private key extraction service is called the Private Key Generator (PKG). The master key in CPK scheme is a matrix in which elements are ECC private keys. The PKG will choose two positive integers w and k as the column count and row count of the matrix. The elements of matrix are randomly generated private keys with ECC domain parameter T. The matrix is denoted as SKM(Secret Key Matrix).

$$SKM = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ r_{21} & r_{22} & \cdots & r_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{w1} & r_{w2} & \cdots & r_{wk} \end{bmatrix}$$

The public domain parameters in CPK are a Public Key Matrix (PKM) derived from SKM. PKM has the same size with SKM, the corresponding elements in SKM and PKM compose a key pair in ECC domain parameters T .

$$PKM = SKM \cdot G = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1k} \\ P_{21} & P_{22} & \cdots & P_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ P_{w1} & P_{w2} & \cdots & P_{wk} \end{bmatrix}$$

$$= \begin{bmatrix} r_{11} \cdot G & r_{12} \cdot G & \cdots & r_{1k} \cdot G \\ r_{21} \cdot G & r_{22} \cdot G & \cdots & r_{2k} \cdot G \\ \vdots & \vdots & \ddots & \vdots \\ r_{w1} \cdot G & r_{w2} \cdot G & \cdots & r_{wk} \cdot G \end{bmatrix}$$

The public parameters also include a cryptography hash algorithm denoted by Map, which maps an arbitrary string into indexes of the matrix used to choose a subset of elements in SKM or PKM. The detail of Map will be described next. So the master key of CPK scheme is SKM and public parameters are {T,w, k,PKM,Map}. In the original scheme, the Map algorithm is implemented though a list of functions satisfying random oracle model. Thus in this paper we simplify the original one to a standard hash algorithm, whose hash length is equal or larger than the required index. The simplified Map algorithm is defined as follows:

Input: Matrix column count w and row count k, hash function H, lH is hash size of H in bits, make sure that $lH \geqslant \lceil log_2^k \rceil$.

Output: $\{s_0, s_1, \cdots, s_w\}$, for every $s_i, 0 \leq i \leq w, 0 \leq s_i < k$

Operation:

- $h \leftarrow H(ID)$.
- i from 0 to w: $s_i \leftarrow h \,\%\, k$, $h \leftarrow \lfloor \frac{h}{k} \rfloor$.

2.3 Key Pair Extraction

In an IBC system, the PKG acts as two roles, first as a authority. When a user registers in the system, he needs to provide some credentials that he has owned the identity. The PKG will generate a private key according to the identity. The private key should be delivered to the user via a secure channel. This can be approached by any methods. In CPK scheme, given an identity, the corresponding private key can be extracted from the private matrix *SKM* and the public key can be extracted from the public matrix *PKM*. Given *ID* is the identity string, *rij*is the element of *SKM* at (*i, j*), *Pij*is the element of *PKM* at (*i, j*). The extraction procedures are as follows:

$$\{s_0, s_1, \cdots, s_w\} \leftarrow Map(ID)$$

$$d = \sum_{i=1}^{w} r_{i,s_i} \; mod \; n, \;\; Q = \sum_{i=1}^{w} P_{i,s_i}$$

From the key generation procedure, we can see that the most notable property of CPK cryptography is its scalability. Since the total number of public keys that can be combined from a*m* × *n* SKM matrix is *mn*. A small 128 × 16 size SKM (or PKM) can combine 1032 number of CPK key pairs. Once the client is preloaded with a 128 × 16 PKM, it does not need to update its information about the whole user space until the total number of user scales up to 1032.While in PKI system, a 1032 number of users necessarily means 1032 number of public key certificate issuing overhead.

2.3.1 Encrypt and Sign

After PKG is established, the master secret will be generated and kept secure in PKG, while the public parameters will be public to every user. A registered user can get his private key from the PKG, the other users' public keys from the public matrix. The key pair is a standard ECC key pair and any standard ECC signature and encryption schemes including ECDSA, ECDH and Elliptic Curve Integrated Encryption Scheme (ECIES) can be used.

2.4 URI Fragment Identifier

Fragment identifier is an optional component of a Uniform Resource Identifier (URI) [7]. As the name implies, it addresses a fragment of the resource denoted by the fragment-free URI. In a URI, the fragment identifier component is indicated by the presence of the first "#" character and terminated by the end of the URI. For example, a URL with a fragment identifier looks like: http: //domain.com/index.html#frag_id. The fragment identifier in a URL is used by the browser to jump to a given portion of the HTML document. Because fragment identifier specified portion is only valid within the context of the main resource, locating and changing the portion neither need to reload the page from the server, nor need to pass the fragment identifier from the browser to the server. So the content of fragment identifier will never appear over the network. This characteristic means that identifier can act as a container to store private information, such as an authentication token or a secret key, for client side web applications. Although fragment identifiers have been used in many web applications, it was first BeamAuth [6] that used the fragment identifier as a security token. WebIBC is inspired by BeamAuth to provide the private key from the fragment identifier to on page JavaScript IBC system.

The browser will retrieve and display a page with URL http://domain.com/index.html#frag_id as follows:

1. Remove the fragment identifier from the URL, use the remaining address http://domain.com/ index.html to retrieve the page. Domain name ("domain.com") and path inside the server ("index. html") will be sent over the network separately to DNS server and the application server.
2. When retrieving the whole page, the browser checks if there exists a portion named by the fragment identifier. If not existed, the browser will ignore the fragment identifier.

3. The downloaded JavaScript in the page can read attributes of the Document Object Model (DOM) object, including the original URL from the attribute "window.document.URL", then it can parse the fragment.

2.5 Working Flow

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

1. The authority trusted by Alice and Bob establishes a PKG, which will generate the system parameters including the public matrix.

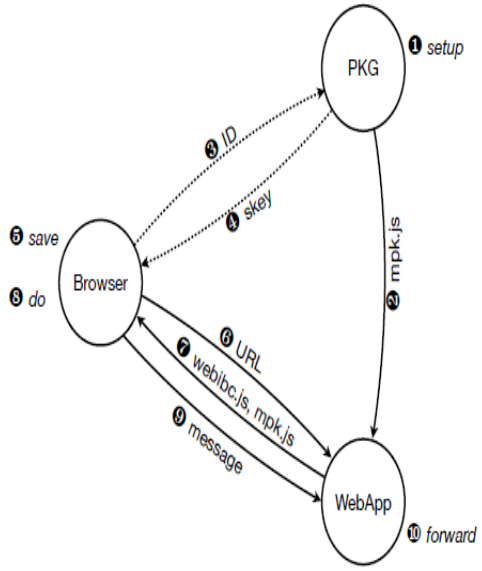

Figure 1. WebIBC Working Flow

2. Web application embeds WebIBC into these systems together with the public system parameters released by the PKG.
3. Alice registers to the PKG with her ID.
4. PKG returns Alice's private key. The communication between users and PKG requires a secure channel for authentication and the transmission of keys. For example, PKG can encrypt the key by a password appointed by Alice and send it to the email address of Alice.
5. Alice 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 Alice 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. Alice uses this web application as normal, entering Bob's email address and message content into the form. When Alice presses the send button, WebIBC JavaScript programs will get the email address from the form, and extract Bob's public key from the matrix, then use this public key with Elliptic Curve Integrated Encryption Scheme (ECIES) to encrypt the message. If Alice wants, she can also generate an ECDSA signature with the private key imported from the bookmark.
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 Bob. Bob 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 Alice.

It should be noticed that all the cryptography operations are all done within the browser, and the server can only receive the ciphertext. The security and privacy of end users can be protected from attacks both on network and server side. From another point of view, server is also free from the burden of cryptography operations which means WebIBC is a good model for distributed computation based on web browsers.

3. IMPLEMENTATION

The WebIBC includes three components: (1) The JavaScript Crypto Library that implements CPK scheme in JavaScript, (2) IBC PKG server that provides private key extraction services, and (3) Web Application Server in which a web mail is provided for the demonstration of concept of WebIBC. In this section, we describe our implementation details of WebIBC. All the system, demonstration and benchmarks, is available at http://infosec. pku.edu.cn/˜guanzhi/webibc/.

3.1 JavaScript Crypto Library

The core function of WebIBC is provided by a JavaScript crpyto library that can be integrated into any web applications. This library includes a set of cryptography algorithms, such as SHA-1, AES, big integer operations, ECC and CPK. We get the SHA-1 and AES JavaScript implementations from the Internet, and Big Integer implementation from a JavaScript RSA library. The BigInteger is designed for RSA, not efficient for ECC, so we implement efficient big integer reduction for NIST primes.

3.2 PKG Server and Web Server

The PKG server acts as a trusted authority in the system. It generates the master key and public system parameters, and provides private key extraction services to users. Before applying WebIBC, a user must authenticate himself to the

PKG; prove the ownership of the identity string, which in our demo system is an email address. If the authentication succeeds, the user can retrieve his private key from the server. For the demonstration, we also provide a sample implementation which is a web PKG server by JavaScript. We do not focus on the security of the server side in this paper, so this sample is just to show how our proposed scheme.

4. SECURITY ANALYSIS

In this section, we will define the attack model and analyze the security of the WebIBC system. In this study, we use CPK algorithm as an IBE algorithm implementation in WebIBC. There are three reasons for us to use this algorithm. First, CPK is an Identity Based Cryptography algorithm that can do both IBE and IBS. Unlike most algorithms such as BF-IBE or PKI based DSA which can only accomplish encrypting or signing task, CPK is an Identity based Cryptography which is able to both encrypt and sign messages. The other reason is that JavaScript is not an efficient language, especially for cryptography algorithm implementations. Fortunately, CPK serves as an efficient algorithm thanks to the fact that it is based on ECC. Furthermore, CPK is remarkable for its scalability.

5. SECURITY OF CRYPTOGRAPHY

The security of WebIBC depends on the security of cryptography algorithms it applies. We will discuss the security of these cryptography algorithms from theory and practice. In WebIBC we apply CPK as our fundamental IBC scheme. As we have discussed, CPK is a bounded IBC scheme, it can defend a fixed number of colluding users. One of the solutions is to replace CPK with other proved secure IBE schemes, for example, BF-IBE has been proved secure in theory. But in current web application environment, these schemes are not efficient enough. For BF-IBE is based on Weil pairing, it needs the computation of at least 512 bit elliptic curve.

6. PRIVATE KEY SECURITY

Modern browsers implement the same-origin policy that prohibits a web object from one site from accessing web objects served from a different site. Browsers currently enforce this by checking whether the two objects' originating domain names, ports and protocols match. However, if the attacker controls the domain mapping, the same-origin policy will be broken. This attack can be accomplished by Trojan horse to tamper the /etc/hosts config file or through the DNS rebinding attack. Then the JavaScript downloaded from the attacker server can gain complete control of the session, including the private key in the fragment identifier. Some recent researches [15, 16] have addressed the attack on same-origin policy but without widespread deployments. So WebIBC is still vulnerable to these attacks.

7. 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 depend son the reliability and correctness of the JavaScript program and data. If the server provides fake script, the security will be broken. We provide some solutions to this problem:

1. WebIBC is a mechanism provided to some honest service provider like Google Mail, to provide a guarantee to customers that they can protect users' privacy by mechanism, not just policy. And because WebIBC is totally "open source", users can check if the provider releases a valid security implementation.
2. In a web page, JavaScript can be linked from other addresses, so the implementation can be downloaded from a trusted server, for example, the server belonging to the PKG.
3. Some browsers have generalized plugins that can replace some contents in a web page, such as CSS or scripts. Users can utilize this kind of plugins to insert the trusted WebIBC implementations into the page.

8. LIMITATION

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

9. CONCLUSIONS AND FUTURE WORKS

In this paper, 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.

REFERENCES

[1] T. Berners-Lee, R. Fielding, and L. Masinter. Uniformresource identifier (URI): General syntex. http://www.ietf.org/rfc/rfc3986.txt, 2005.

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

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

[4] X. Boyen. General ad hoc encryption from exponent inversion ibe. In LNCS 4515, Springer- Verlag, pages 394–411, 2007.
[5] B.Schneier. Applied cryptography: Protocols, algorithms, and source code in c, second edition. 1996.
[6] C. Cocks. An identity based encryption scheme based on quadratic residues. Lecture Notes In Computer Science, 2260:360–363, 2001.
[7] D. Hankerson, A. Menezes, and S. Vanstone. Guide to elliptic curve cryptography. Springer-Verlag, 2004.
[8] F. Hess. Efficient identity based signature schemes based on pairings. In SAC 2002, LNCS 2595, Springer-Verlag, pages 310–324, 2003.
[9] C. Jackson, A. Barth, A. Bortz, W. Shao, and D. Boneh.Protecting browsers from dns rebinding attack. ACM Conferenceon Computer and Communications Security, 2007.
[10] C. Karlof, J.D.Tygar, D. Wagner, and U. Shankar. Dynamicpharming attacks and locked same-orgin policies for webbrowsers. ACM Conference on Computer and CommunicationsSecurity, 2007.
[11] K. Paterson. ID-based signatures from pairings on ellipticcurves, cryptology eprint archive, report 2002/004. http://citeseer.ist.psu.edu/paterson02idbased.html.
[12] A. Shamir. Identity-based cryptosystems and signatureschemes. Crypto ’84, pages 47– 53, 1985.

BIOGRAPHIE

Archana B. Kadga from Bidar Karnataka state (INDIA), had completed B.E.(CSE) in Basavakalyan Engineering College, Basavakalyan and M.Tech (CSE) in Visvesvarayya Technological University “JNANA SANGAM” Belgaum. Now she is working as Assistant Professor in Computer Science Department at Sahakar Maharshi Shankarrao Mohite Patil Institute of Technology and Research Center, Akluj Solapur District (Maharastra). Her area of interests are Cryptography, Security ,Networking Field and Computer Applications.