ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Cryptography Based Solutions for Web IBC Security

R.Karthikeyan[1], V.Khanna[2]

[1]Assitant Professor, Dept. of CSE, Bharath University, Chennai, Tamil Nadu, India

[2]Dean, Dept. of IT, Bharath University, Chennai, Tamil Nadu, India

ABSTRACT: In the current scenario the web applications are growing rapidly which led users to give their data to online application providers. These activities will endanger the security and privacy of the users. A new approach using webIBC was introduced , which integrates public key cryptography into web applications without any use of browser plugins. The identity based cryptography provides the public key of webIBC by eliminating the need of public key and certificate online retrieval. Also it has the implementation and performance evaluation details of webIBC. Hence, webIBC provides greater security for the client-side in web applications.

I. INTRODUCTION

Now-a-days 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. 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. 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.

II. IMPLEMENTATION OF STANDARDS

Public key cryptography based solutions for the desktop counterpart of the above web applications have been deployed widely for many years. PGP and S/MIME are two de facto standards, and have been implemented within applications inside many desktop mail clients. The key management of these solutions requires 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.

2.1 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,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

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 key into 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 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.

2.2 Resolving challenges:

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 and lynx. 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.

III. WEBIBC BASIS

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.

3.1 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. After the concept was first suggested, IBS schemes were sooner founded, but IBE scheme remained a more challenging problem. Until 2001, the Boneh-Franklin IBE (BF-IBE) based onWeil 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.

3.2 Combined Public Key

The CPK cryptosystem is based on the elliptic curve cryptography (ECC). Let the elliptic curve domain parameters discussed in this paper be a sextuple: T = (p, a, b,G, n, h) consisting of an integer p specifying the finite field Fp, two element a, b $\in$ Fp specifying an elliptic curve E(Fp) defined by the equation E : $y^2 \equiv x^3 + a.x + b \pmod{p}$, a base point G = (xG, yG) on E(Fp), a prime n which is the order of G, and an integer h which is the cofactor h = E(Fp)/n. The ECC key pair (d, Q) associated with T consists of an elliptic curve secret key d which is an integer in the interval [1, n - 1], and an elliptic curve public key Q = (xQ, yQ) which is the point Q = dG. One character of ECC key pair is that the combination of private keys and corresponding public keys in ECC is still a pair of elliptic curve keys. For example, s1, s2 are private keys in elliptic curve, the corresponding public keys are Q1, Q2 that Q1 = s1 · G, Q2 = s2 · G; s = s1 +

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

s2, Qt = st · G = (s1 + s2) · G = (s1 · G) + (s2 · G) = Q1 + Q2. So the combination of key pairs will result into a new key pair.

3.3 System Setup

In the setup procedure, a trusted authority will generate a master secret in the system and public parameters known 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 .
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.

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

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

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

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 $\geq \left| \log_2^k \right|$

Output: {s0, s1, · · · , sw}, for every s_i , $0 \leq i \leq w$, $0 \leq s_i < k$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Operation:

• h ← H (ID).

• i from 0 to w: si ← h % k, h ← $\left\lfloor \frac{h}{k} \right\rfloor$

3.4 Key Pair Extraction

In an IBC system, the PKG acts as two roles, first as an 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:

{s0 s1, · · · , sw} ← Map (ID)

d = $\sum_{i=1}^{w} ri, si$ mod n, Q= $\sum_{i=1}^{w} Pi, si$

And d is the private key, Q is the public key, fromsection 2.2, it is proved that (d, Q) is also an ECC key pair. 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.

3.5 URI Fragment Identifier

Fragment identifier is an optional component of a Uniform Resource Identifier (URI). 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 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 identifier string from the original URL string, and use it for any propose.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

3.6 BC's 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.

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 a 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.

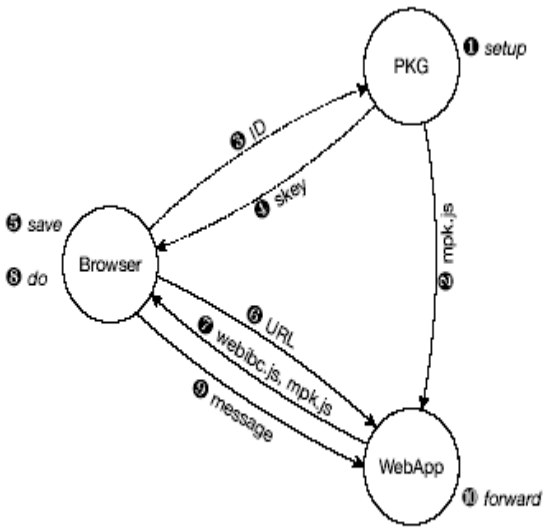

Figure 1. WebIBC Working Flow

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 URL fragment identifier component.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

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 cipher text. 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.

IV. 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/.

4.1 JavaScript Crypto Library

The core function of WebIBC is provided by a JavaScript crypto 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 Big Integer is designed for RSA, not efficient for ECC, so we implement efficient big integer reduction for NIST primes.

4.2 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 works. We implement a simple web mail as our application Server. After login, users can send messages and view received messages.

V. PERFORMANCES AND EVALUATION

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 effi- ciency 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.

5. 1 Computation Overhead

Computation overhead is determined by many factors, including the performance of testing environment, how much data to process, how many recipients are involved and the required security level of cryptography schemes. Before the evaluation, two questions must be answered; the first is what are the average values of these factors for a given application? The other thing is, what is the maximum overhead the application can sustain without destroying user experiences. For a practical mechanism, answer to the second question must be satisfied, while how to satisfy the above factors is a tradeoff that can be the design choice of the system. We select distinct machines for the test. Our main benchmark environment is an Apple Mac Book laptop with 1.83 GHz Intel Core 2 Duo processor, 2 GB RAM and installed with dual operating systems. The JavaScript programs are tested in Safari 3.03, Fire fox 2.0.0.9, Opera 9.24 on

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Mac OS X and Internet Explorer 7 (IE7) on Windows XP Professional. We also test the system on a much older Dell desktop computer with 2.6 GHz Pentium 4 processor and 256 MB RAM running Internet Explorer 6 (IE6). From some tests the answer to the second question is found out. For all the four browsers there is a timeout mechanism on the JavaScript programs. When a JavaScript program running over about 5 seconds without any response, the browser will consider the program to be a "Slow Scrip", and hence pause it and display an alert dialog to ask the user whether to terminate it or not. The 5 seconds limit is not affected by the speed of the computer; it remains the same in all our test machines. This alert will break the running and confuse the user, so 5 seconds may be considered as the upper bound of computation time overhead. The computation of WebIBC is mainly coming from three basic cryptography building blocks: AES, SHA-1

Table 1. Benchmarks for the computation overhead of the three basic cryptography building blocks

	IE	Safari	Firefox	Opera
SHA-1	20.5	22.1	39.4	25.1
AES	13. 46	14.6	15.4	8.4
MUL	1.12	2.09	1.91	1.39

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 ofWebIBC 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. Tab.1 shows the result of test on the unit speed of these functions. Now let us consider the factors affecting the final result:

ECC Key Length Every CPK operation requires k elliptic curve point addition and one scalar multiply. The details of this algorithm will not be introduced here, the descriptions will be found in [13]. One point addition includes 10 MULs; one point doubling includes 8 MULs. Given l is the key length in bits, there needs l point doublings and 0.5l point additions on average. The selected key length is 192 bits, providing security the same as 1776 bits RSA. This requires about 2500 MUL operations for a CPK encryption.

Payload Size The average data size is different for various applications. For example the most popular web mail, email statistics from UC Berkeley [4] show that the average email content size is 1863 bytes. So we use three different testing data sizes, 0.5KB, 2KB and 10KB. In our test we did not consider the process of email attachments, for one reason is attachment might be several megabytes which is too large for current JavaScript cryptography functions to process, the other is we did not find a method for JavaScript to get the attachment before it is sent to the server through a HTTP POST.

The result of our test is shown in Fig.2. Even on the slowest browser with the biggest data size, the total time is from 2.1 seconds to 3.6 seconds, less than the 5 seconds limitation. For the average data size - 2KB and the three of the most popular browsers IE, Fire fox and Safari, the overhead is about the same, from 1.4 seconds to 1.6 seconds.

5.2Possible Optimization

Our tests on the slower computer will exceed the 5 seconds limit. So WebIBC still needs some optimization to fulfill the need of old machines. For the implementation of WebIBC, some possible optimization methods are listed here.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

	0.5KB	2KB	10KB
Safari	1383.7	1,492	2,071
Firefox	1,523	1,661	2,401
IE	1,459	1,698	2,791
Opera	2,110	2,349	3,628

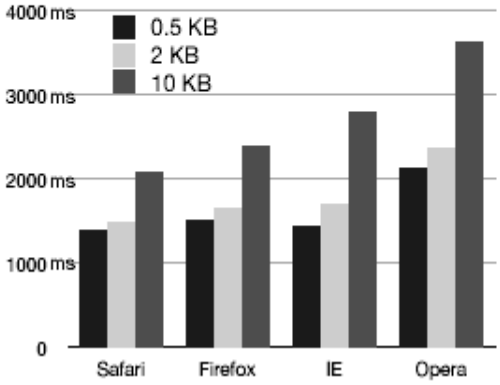

Figure 2. Evaluation Result of Computational Overhead

1. 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.[1]

2. Faster ECC algorithm. In our proof of concept implementation, the slowest point scalar multiply algorithm is used. In [13] some optimized algorithms are introduced. And for encrypt and sign operations, 66% of the scalar multiply can be pre-computed.[2]

3. Technique methods. 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[3]. 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.

5.3 Bandwidth Overhead

The bandwidth overhead caused by WebIBC is coming from the downloading of JavaScript crypto library and system public parameters of IBC scheme. In our prototype, the crypto library includes AES, SHA-1, big integer operation, ECC and CPK JavaScript implementations. Current total WebIBC JavaScript code size is 61KB, about 1600 lines. The size of public system parameters is determined by the size of the CPK public matrix. Given c is the matrix column count and the r is row count, it requires about 96 × c × r
bytes to store the public matrix with uncompressed elliptic curve points, or about 49×c×r bytes with compressed elliptic curve points. In a 32×32 matrix with 192 bits ECC, the size is less than 100KB. Since the most remarkable property of CPK is its scalability we have mentioned about, this 100KB public matrix can combine up to 3232 number of CPK key pairs. The format of elliptic curve points is a tradeoff in WebIBC. Although the compressed format decreases half of the matrix size, there needs r extra computations to get the y coordinate of the mapped elliptic curve points. From our last test, it shows that at current state the computation is the bottleneck of our system, because compared to a web page, the bandwidth overhead is acceptable, and if the browser has cached on these JavaScript files, it can be neglected.

5.4 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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

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. As specified in Section.2.2.2, a small size of key parameters may serve for
a large amount of users, which is a salient property fitting large-scale web applications.[4]

5.4.1 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.InWebIBC 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 onWeil pairing, it needs the computation of at least 512 bit elliptic curve.[5] Currently we do not have an implementation of BF-IBE on JavaScript, from a benchmark of ECC performance, the 512 bit elliptic curve is ten times slower than the 160 bit elliptic curve, and the ECC point scalar multiply is only a low level operation. So currently, BF-IBE is not practical. Cocks IBE [12] is also not efficient with respect to bandwidth, for every bit of plaintext, the output cipher text will be expanded to 2048 bit. The security of cryptographic hash algorithm should also be considered. The hash algorithm will be used in two places, the CPK map function and the digital signature algorithm. In our implementation we use SHA-1. While SHA-1 has recently been shown to have certain weaknesses, these weaknesses can be resolved by replacing SHA-1 with stronger SHA-2 family hash algorithms[6]. It is the observation of that the asymmetric key cryptography is approximately 100 times slower than the symmetric key cryptography. So the computation overhead of hash algorithm in WebIBC for average email data size is negligible. We will present further results to support this argument as future work.

5.5 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.[7] 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 have addressed the attack on same-origin policy but without widespread deployments. So WebIBC is still vulnerable to these attacks.[8]

5.5.1 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, phishers often convince recipients to respond. Technical subterfuge schemes plant crime ware onto PCs to steal credentials directly, often using Trojan key logger spy ware. Pharming crime ware misdirects users to fraudulent sites or proxy servers, typically through DNS hijacking or poisoning[9].

5.5.2. Server Cheating

One possible threat is coming from the application server.[10] 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[11]. 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[12].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

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.[13]

5.5Limitation

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.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we present Web IBC 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 Web IBC is resilient to some known attacks using the proposed schemes. The future work ofWebIBC is to evaluate the feasibility of other IBC schemes on WebIBC, especially BF-IBE.

REFERENCES

[1] CNN news: Yahoo accused of misleading congress.http://www.cnn.com/2007/US/10/16/yahoo.congress/.
[2] Kaliyamurthie K.P., Parameswari D., Udayakumar R., "QOS aware privacy preserving location monitoring in wireless sensor network", Indian Journal of Science and Technology, ISSN : 0974-6846, 6(S5) (2013) pp.4648-4652.
[3] Epic online guide to practical privacy tools.http://www.epic.org/privacy/tools.html.
[4] Lynx: a text browser for the world wide web.http://lynx.browser.org/.
[5] Sharmila D., Muthusamy P., "Removal of heavy metal from industrial effluent using bio adsorbents (Camellia sinensis)", Journal of Chemical and Pharmaceutical Research, ISSN : 0975 – 7384, 5(2) (2013) pp.10-13.
[6] UC Berkerly email stats.http://www2.sims.berkeley.edu/research/projects/howmuch-info/internet/.
[7] W3m, a text-based browser and pager.http://w3m.sourceforge.net/.
[8] Udayakumar R., Khanaa V., Saravanan T., Saritha G., "Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction", Middle - East Journal of Scientific Research, ISSN : 1990-9233, 16(12) (2013) pp.1781-1785.
[9] B. Adida. Beamauth: two-factor web authentication with abookmark. In CCS '07: Proceedings of the 14th ACM conference on Computer and communications security, pages48–57, New York, NY, USA, 2007. ACM.
[10] Kalaiselvi V.S., Prabhu K., Ramesh M., Venkatesan V., "The association of serum osteocalcin with the bone mineral density in post menopausal women",Journal of Clinical and Diagnostic Research, ISSN : 0973 - 709X, 7(5) (2013) pp.814-816.
[11] T. Berners-Lee, R. Fielding, and L. Masinter. Uniformresource identifier (URI): General syntex.http://www.ietf.org/rfc/rfc3986.txt, 2005.
[12] Jayalakshmi T., Krishnamoorthy P., Kumar G.R., Sivamani P., "The microbiological quality of fruit containing soft drinks from Chennai", Journal of Chemical and Pharmaceutical Research, ISSN : 0975 – 7384, 3(6) (2011) pp. 626-630.
[13] D. Boneh and X. Boyen. Secure identity based encryptionwithout random oracles. Proceedings of Crypto 2004,LNCS. Springer-Verlag, 2004.
[14]Dr.A.MuthuKumaravel, KNOWLEDGE BASED WEB SERVICE, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801,pp 5881-5888, Vol. 2, Issue 9, September 2014
[15]Dr.A.Muthu Kumaravel, Data Representation in web portals, International Journal of Innovative Research in Computerand Communication Engineering, ISSN(Online): 2320-9801,pp 5693-5699, Vol. 2, Issue 9, September 2014
[16]Dr.Kathir.Viswalingam,Mr.G.Ayyappan,A Victimization Optical Back Propagation Technique in Content Based Mostly Spam Filtering ,International Journal of Innovative Research in Computerand Communication Engineering ,ISSN(Online): 2320-9801 , pp 7279-7283, Vol. 2, Issue 12, December 2014
[17]KannanSubramanian,FACE RECOGNITION USINGEIGENFACE AND SUPPORT VECTOR MACHINE,International Journal of Innovative Research in Computerand Communication Engineering,ISSN(Online): 2320-9801,pp 4974-4980, Vol. 2, Issue 7, July 2014.
[18]Vinothlakshmi.S,To Provide Security & Integrity for StorageServices in Cloud Computing ,International Journal of Innovative Research in Computer and Communication Engineering ,ISSN(Online): 2320-9801 , pp 2381-2385 ,Volume 1, Issue 10, December 2013