Executive Summary of Minor Research Project on

SECURED DATA TRANSACTION IN A NETWORK USING ITENTITY BASED CRYPTOGRAPHY

Principal Investigator

Mr.Prakash Kumar.P
Department of Computer Science
Vivekananda College,Puttur

Purpose

The purpose of this research project is to demonstrate that traditional Transport Layer Security can be replaced with identity-based encryption to eliminate the need of site certificates. Data for the proof-of-concept is given.

Secure communication is an intrinsic requirement of today's world of on-line transactions. Whether exchanging financial, business or personal information, people want to know with whom they are communicating (authentication) and they wish to ensure that the information is neither modified (data integrity) nor disclosed (confidentiality) in transit. 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 report I present **NetIBC** project which integrates public key cryptography into web applications without any browser plugins. The implementation and performance evaluation demonstrate that **NetIBC** is secure and efficient both in theory and practice.

An Identity Base Encryption (IBE) scheme is a public-key cryptosystem where any string is a valid public key. In particular, email addresses and dates can be public keys. For many situations in distributed network environments, Identity Base cryptography is a must during communications.

Requirements for security in information exchange generally have three aspects:

- **Confidentiality**: Only the intended recipient of a message has access to the message content.
- **Integrity**: The message is received without modification.
- **Authenticity**: The origin of the message can be verified.

NetIBC lets you easily address all three aspects

The Problem Definition

Suppose that someone wants to buy a product from an online retailer. With traditional SSL, the web browser connects to the server and downloads the certificate. It checks to see that it has been signed by a trusted party, and extracts the public key of the server. It then encrypts data with the public key and sends it to the server, which decrypts the data and completes the transaction. The main disadvantage to this scheme is that certificates rely on the user to manage them, and most web users are not even aware of SSL technology, much less certificates. Certificates can also be revoked, but most users do not check the status of the certificate before transferring secure data.

If SSL used the IBE system instead, there would only be the need for one certificate: the master certificate embedded in web browsers for the PKG. The browser would then encrypt data with the URL of the website and send the encrypted text to the server. The server would get the encrypted message from the client and decrypt it with its private key that it has received from the PKG. There is no need for the server to have its own certificate: it merely needs the private key generated from the PKG to decrypt any message sent to it. This process is shown graphically in Figure 1.2

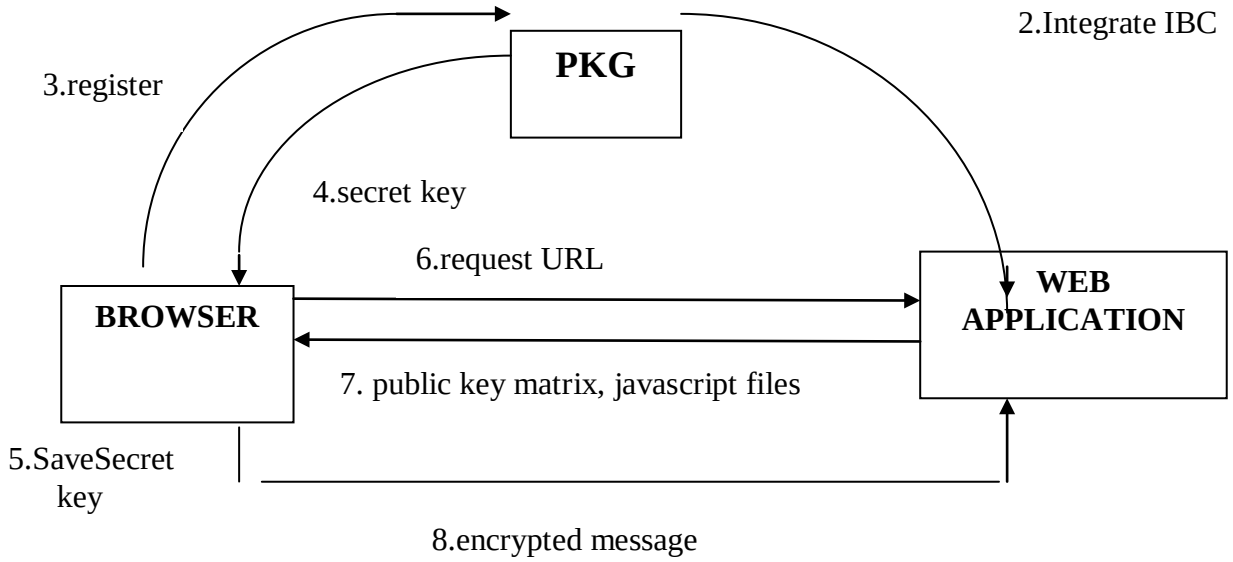


Figure 1.2 Workflow diagram

Scope

NetIBC enhanced with web Identity Based Cryptography have the following features:

- When sending email using Identity based cryptography there is no need for an online lookup to obtain the recipient's certificate.
- Senders can send email that can only be read at some specified time in the future, since public key contains expiration date.

Public key certificates contain a preset expiration date. In an IBE system key expiration can be done by having user1 encrypt e-mail sent to user2 using the public key:

User2@u2.com||current-year".

In doing so user2 can use his private key during the current year only. Once a year user2 needs to obtain a new private key from the PKG.

Algorithms used in Identity Based Cryptography:

- ✓ System setup
  - ➢ Generates Master secret and Public Parameters
- ✓ Extraction
  - ➢ Obtaining Public key and private key from Secret Key Matrix (SKM) using Elliptical curve concept
- ✓ Encrypt
  - ➢ Encrypting with the Public key.
- ✓ Decrypt
  - ➢ Decrypting with the private key.

System Setup and Implementation

Challenging Tasks

With the growing approval of Web applications like Facebook ,Gmail , google cloud etc, 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 technology these applications only rely on browsers with common features including HTML, JavaScript and CSS, without the need of installing any browser 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. 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.

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 key into JavaScript programs. For some solutions, a program 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.

In this research project(NetIBC) , two mechanisms are combined to decide the above challenges and provide security and privacy for client side. The first one is Identity Based Cryptography (IBC), a type of public key cryptography in which the public key can be an arbitrary string. **NetIBC** 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 Hyper Text form in the message sending web page. I have used JavaScript in implementing the project which incorporated into any web applications, and run in all browsers (I have tested it in Internet Explorer , Chrome and Firefox) . The other is to provide the private key from the URL fragment identifier, the substring starting from the first "#" symbol of a URL. In this project the private key is encoded into the fragment identifier component of the web application URL.

Key Generation

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

$E : y^2 \equiv x^3 + a \cdot x + b \pmod p$, 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 $h = E(\mathrm{F}p)/n$. 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

$Q1 = s1 \cdot G, \quad Q2 = s2 \cdot G \quad s = s1 + s2,$

$Qt = st \cdot G = (s1 + s2) \cdot G = (s1 \cdot G) + (s2 \cdot G) = Q1 + Q2$

So the combination of key pairs will result into a new key pair.

System Setup (Authentication and Key Generation)

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 I and j 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) used for key generation

$SKM =$

$$\begin{pmatrix} r11 & r12 & \cdots & r1j \\ r21 & r22 & \dots & r2j \\ \vdots & \dots & \ddots & \vdots \\ ri1 & ri2 & \cdots & rij \end{pmatrix}$$

The public domain parameters Public Key Matrix (PKM) derived from Secret Key Matrix. PKM and SKM are of same size and in my project I have selected finite matrix of size 32x32 .The corresponding elements in SKM and PKM compose a key pair in ECC domain parameters T

$$PKM=SKM.G= \begin{pmatrix} p11 & p12 & \cdots & p1j \\ p21 & p22 & \dots & p2j \\ \vdots & \dots & \ddots & \vdots \\ pi1 & pi2 & \cdots & pij \end{pmatrix} = \begin{pmatrix} r11.G & r12.G & \cdots & r1j.G \\ r21.G & r22.G & \dots & r2j.G \\ \vdots & \dots & \ddots & \vdots \\ ri1.G & ri2.G & \cdots & rij.G \end{pmatrix}$$

The mapping from identity to the coordinates of the matrix is implemented through a hash function. The output of HASH will be adjusted to 190 bits, denoted Y S, in which every 5 bits is denoted by w0,w1, . . . ,w37, wi can be converted to an integer in [1, 32].

$$YS = HASH(ID) = w0,w1, \ldots ,w37$$

The w0 is used to determine the initial coordinate of the column, the following column coordinate is the result of adding1 to the above column coordinate. w1 to w32 is used to indicate the raw coordinates, and w33 to w37 is used to indicate the System Key coordinate.

Key Extraction

When a user registers in the system, he needsto 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:

$\{s0, s1, \cdots, sw\} \leftarrow Map(ID)$

d= $\sum_{i=1}^{w} r_i s_i$ mod n Q=$\sum_{i=1}^{w} P_i s_i$

here *d* is the private key,*Q* is the public key

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 can be used.

Workflow

Suppose User1 wants to send a secure message to User2 throw a Secure mail improved by NetIBC using IBC Then work flow is as follows

1. The authority trusted by User1 and User2 establishes a PKG(Private Key Generator), which will generate the system parameters including the public matrix.
2. Web application embeds NetIBC into these systems together with the public system parameters released by the PKG.
3. User1 registers to the PKG with her ID.
4. PKG returns user1'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 user1 and send it to the email address of Alice.
5. user1 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 user1 can use this bookmark to log into the web application.

7. The NetIBC JavaScript files will also be downloaded from the server, including the public matrix of system.
8. User1 uses this web application as normal, entering User2's email address and message content into the form. When User1 presses the send button, NetIBC JavaScript programs will get the email address from the form, and extract User2's public key from the matrix, then use this public key with Elliptic Curve Integrated Encryption Scheme (ECIES) to encrypt the message.
9. And then the message will be sent to the server
10. Because the message has been protected, the Web application can not modify the message but only forward it to User2. He 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 user1.

Implementation Details

NetIBC project mainly contains four components. Most of the code is implemented using Java Script. Main reason to select java script is its popularity and almost all browser's supports java script applications. The modules are described below

NetIBC Modules:

- Authentication module
- Encryption Module
- KEY Server
- Decryption Module

Authentication module

This module will perform the authentication process. Every user must register to the E-Mail server as well as Key-Server. The E-Mail server will issue the Login Details and the key-Server will generate the public and private key for each and every user. Thus it allows only authorized users to access our E-Mail server.

Encryption Module

This module is useful in achieving the security for our whole system by encrypting the E-Mail with DESede algorithm before sending the mail to the E-Mail server. Thus server will receive only the cipher text , because the E-mail message is encrypted in the client browser itself.

Key Server Module

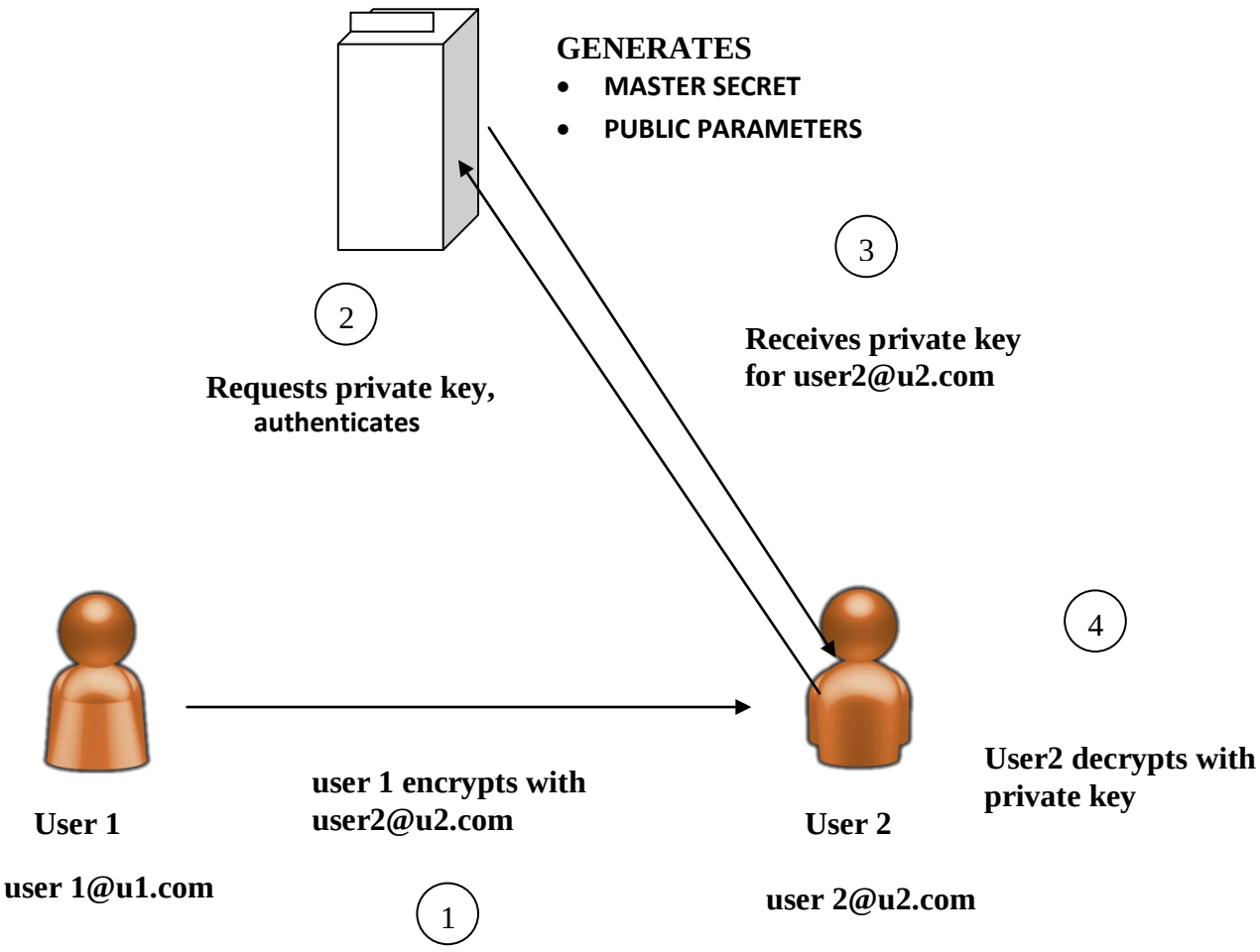
The Key-Server module will generate the Private Key for each and every user and it will issue the private key to the users those who want to read the mail. Before issuing the private key to the user it will verify the user weather he is an authorized user or not.

Decryption Module:

The decryption module will decrypt the Received mail by using the downloaded private key from key-Server. After decrypting the mail the user can able to read that mail, unless he can't read the Mail.

1. Establishing Trusted Private Key Generator Setup.
2. Web application embeds NetIBC into these systems together with the public system parameters released by the PKG.
3. User registers to the PKG with their Identity
4. obtain the private key from PKG
5. Save the private key along with the URL as book mark
6. Use the bookmark to access the web application.
7. NetIBC javascript files and public key matrix are downloaded from the server.
8. The mail is encrypted using the receiver's public key from public key matrix using ***"Elliptic Curve Cryptography"*** and the mail is sent to the server.

   The System architecture is given in figure 2



SYSTEM TESTING

The purpose of testing is to discover logical errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. I have conducted various test procedures in three main browsers (Internet Explorer , Firefox and Google chrome)

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. All components tested in these testing phase given desirable results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide a systematic demonstration that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.
encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

The systematic Testing of all test cases mentioned above passed successfully. Defects found in some procedures are corrected and final test is conducted without any defects encountered.

Overall Analysis

Result Analysis

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

To test the efficiency of the system, I encrypted and decrypted sample files using the IBE system and measured the benchmark times for the encryption and decryption stages. . Initially I ran the client and server programs on the same machine to eliminate the possibility of interval between the two programs which could affect results in the initialization phase. The encryption phase provided more results than the decryption phase benchmarks, but in both cases the total time was provided. The Content size did not seem to affect the processing of the file, and indeed the decryption time did go down with the larger data. Realistically, this time should increase with much larger files, but in an actual IBC system, the user would not be transmitting much more than credit card information and mailing

addresses, so the total amount of information sent would be relatively low. The only potential problem in this scheme is that the computer takes more time to decrypt a file . It is possible that servers running at e-commerce sites would have much more processing power than my test server and could certainly handle more secure connections per second. All testing procedures are incorporated

Computational Overhead

Computational load 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 main things should be considered -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.

Practically Second should be satisfied , I have selected initially one machine (Windows OS) for test. Later I have used distinct machines including Linux (Ubuntu 10.04) . The main benchmark environment is an Windows Xp 2.4GHz GHz Intel Duo processor, 2 GB RAM and installed The JavaScript programs are tested in Internet Explorer 6, Firefox 3.0 , Google chrome and Internet Explorer 7 (IE7) on Windows XP Professional.

For all the browsers used , I have written timeout functions in java script. When browser running the java script code for more than 5 Seconds without any response , A alert message is displayed informing that browser is slow. The 5 seconds limit is not affected by the speed of the computer; it remains the same in all my tests. This alert will break the running . 5 seconds may be considered as these upper bound of computation time

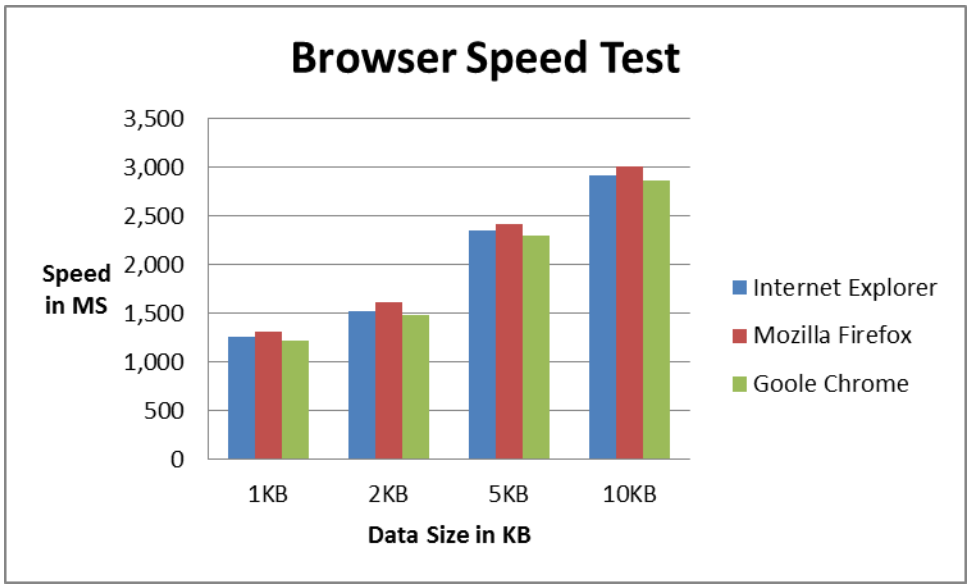
The computation load is mainly coming from three basic cryptography building blocks: AES, SHA-1 and integer module Multiply (MUL) operation. The MUL is the main computation component of ECC and CPK. When the unit performance is known, the total running time can be estimated for specific factors.

ECC Key Length Every CPK operation requires *k* elliptic curve point addition and one scalar multiply. The details of these are already discussed in the Chapter 3,. One point addition includes 10 MULs, one point doubling includes 8 MULs. Given *len* is the key length in bits, there needs *len* point doublings and 0.5*len* point additions on average. The selected key length is 192 bits, providing security the same as 1776 bits RSA. This requires about 2500MUL operations for a CPK encryption.

The average data size is different for various applications. Most popular webmail, email statistics from UC Berkeley Which show that the average email content size is 1863 bytes. So I have used four different testing data sizes, 0.5KB, 2KB, 5KB and 10KB. In the test I did not consider the attached files (Which may very large for my program or function to handle.

The final result of test is shown in table below . Even on the slowest browser with the biggest data size, the total time is found to be 2.3 Seconds to 3.9 Seconds which is less than upper bound selected (which is 5 seconds) .For the average data size - 5KB and the e most popular browsers IE, Firefox and Google chrome, the overhead is about the same, from 2.4 seconds to 2.6 seconds.

Data Size(	Browsers Tested (in Milliseconds')		
	Internet Explorer	Mozilla Firefox	Goole Chrome
1KB	1,254	1,310	1,215
2KB	1,527	1,611	1,478
5KB	2,354	2,421	2,301
10KB	2,916	3,014	2,868



Bandwidth Considerations

Other overload to be considered is the overhead caused by the system from the downloading of JavaScript library and system public parameters of IBC scheme. Since Elliptical Cryptographic operations include big integer arithmetic ,Java Script code size is around 70KB of about 2000 lines. Given c is the matrix column count and the r is row count, it requires about $96 \times c \times r$ bytes to store the public matrix with uncompressed elliptic curve points, or about $49 \times c \times r$ bytes with compressed elliptic curve points. In a 32×32 matrix with 192 bits ECC, the size is less than 100KB. 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. the bandwidth overhead is acceptable, and if the browser has cached on these JavaScript files, it is negligible.

Security Analysis-Private Key

In this section, various attacks are considered and analysed the security of the system. The security of System depends on the security of cryptography algorithms it applies. I will discuss the security of these cryptography algorithms from theory and practice. InIn this project I apply CPK as our fundamental IBC scheme. As we have discussed, CPK is

a limited IBC scheme, it can defend a fixed number of conspiring 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. Currently we do not have an implementation of BF-IBE on this project , 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.
The security of cryptographic hash algorithm should also be considered. The hash algorithm is 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. So the computation overhead of hash algorithm in this project for average email data size is negligible.

Current 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. have addressed the attack on same-origin policy but without widespread deployments. So I can not guarantee that this application is free attacks. It can say that its is at risk to some types of attacks.

Summary

In the discussion above, I have shown that for a cryptosystem to effectively operate as a secure network public-key cryptographic protocol, it must have certain properties. Of course, the underlying cryptosystem must be secure. I have shown the system implemented has all of the desired properties for a secure public-key cryptosystem. I have also shown how the

IBC system differs from RSA in terms of its mathematical properties. The system must also exhibit some features for it to work similarly to SSL and I have outlined these features also. The protocol must first have the ability to create public and private keys that are mathematically hard to break. The protocol must be able to quickly generate private keys for servers and have a secure method of distributing the private key to the server, and finally be able to decrypt messages sent to it based on its ID. I have shown that these features are easily implemented .

Conclusion

In this report, I have shown that a complete scheme for secure network transactions can be accomplished without the need for certificates. I have shown that an alternative cryptosystem to the current RSA system has comparable security and similar ease of use.

Future Enhancements

In this report, I have presented only a simple implementation of a complete working Web IBC system. There are numerous modifications that could be made to this project in future designs. For example keys can be given time frame after which cause keys to automatically expire. I have tested this project only on windows platform and Linux Platforms only .It could be implemented in other platforms like MAC and Android Systems . Also I have applied cryptography only for text data and File attachments are not considered. It can be enhanced to encrypt Graphical images.