ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012

Identify Basis Cryptography for system bounding Security in URL Applications

[1]K.Ramakrishna, [2]B.Srinivasulu, [3]U.Rakesh, [4]B.SatheeshKumar, [5]M.Srinivasa Rao

M.Tech (SE), Dept. of CSE, Holy Mary Institute Of Technology And Science,Hyderabad,India.

M.Tech (CSE), Dept. of CSE, Holy Mary Institute Of Technology And Science,Hyderabad,India.

M.Tech, Dept. of CSE, Holy Mary Institute Of Technology And Science,Hyderabad,India.

Dept. of CSE, Holy Mary Institute Of Technology And Science,Hyderabad,India.

Dept. of CSE, Holy Mary Institute Of Technology And Science,Hyderabad,India.

Abstract – This paper we propose and evaluate we present, which integrates public key cryptography into url applications without any browser plugins. The public key of is provided by identify basis cryptography, eliminating the need of public key and certificate online retrieval; the private key is supplied by the fragment identifier of the URL. In , two mechanisms are integrated to resolve the above challenges and provide security and privacy for system bounding url users. The first one is Identify Basis Cryptography (IBC), a type of public key cryptography in which the public key can be an arbitrary string. can provide public key encryption and digital signature for the url 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 url page, the implementation of IBC can make easily integrated into any url applications, and run in all browsers even text basis http systems. The other is to provide the private key from the URL fragment identifier. In , the private key is encoded into the fragment identifier component of the url application URL.

Keywords: True Random Number Generator , Pseudo-random Number Generator , Key Pair Extraction, Fragment Identifier, Cryptography.

I. INTRODUCTION

Two mechanisms are integrated to resolve the challenges and provide security and privacy for system bounding url users. The first one is Identify Basis Cryptography (IBC), a type of public key cryptography in which the public key can be an arbitrary string. can provide public key encryption and digital signature for the url 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 url page, the implementation of IBC can make easily integrated into any url applications,

and run in all browsers even text basis http systems. The other is to provide the private key from the URL fragment identifier. In , the private key is encoded into the fragment identifier component of the url application URL.

Existing System:

These online applications offer reliable storages and ease to access services. With the latest techniques these applications only rely on browsers with common features 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. Public key cryptography basis solutions for the desktop counterpart of the above url applications have been deployed widely for many years. Many applications have been implemented within applications inbounding many desktop mail systems. The key management of these solutions requires ad-hoc trust management centralized Public Key Infrastructure (PKI). Generally, these methods can be classified into desktop software and browser plugins. 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 url applications.

II.SYSTEM OVERVIEW

A.Related Researches and Reports

In spite of many advantages the identify-basis cryptosystem provides over traditional public key basis cryptosystem, the paradigm requires frequently user authentication and secure channel for private key issue, which has handicapped its wide acceptance and restrict its usage to a small and closed groups where a central trusted authority exists and is easily accessible. In this paper1 we propose a framework basis on the Trusted Computing (TC) techniques to improve the efficiency of private key issue in identify-basis cryptosystem. We take the Trusted Platform Module (TPM) as a local trusted authority for key extraction. The model, scheme and a survey on how to implement popular identify-basis key issue on TPM are given. The security and performance analysis are provided, together with implementation issues for several popular identify-basis cryptographic schemes.

Identify-Basis Cryptography (IBC) is a form of public key cryptography for which the public key can be an arbitrary string, such as email address, phone number or other user's identify information. The concept was first introduced to eliminate the complexity of certificate management. The truth is no secure random number generator is available for Url basis applications. Random number generator is one of the most fundamental primitives in cryptography that has been researched for many years. "A random number generator is a device or algorithm which outputs a sequence of statistically independent and unbiased binary digits.True randomness is widely used in cryptography applications, such as symmetric and asymmetric cryptography key generation. Weak random numbers may offer the adversary abilities to bypass the hardness of breaking a cryptosystem. However, in spite of the importance of random number generation security, many designs, standards and protocols used in practice instead leave the random number generator to non-security exports, many real world implementations only rely their security on insecure solutions.

The most recent example is a random number generator defect found in Debian Linux . This flaw results in a large amount of security applications include SSH.Therefore, current Url basis security applications even without a random number generator will result in great danger. In this paper we describe the design and implementation of a random number generator for Url basis security applications. The security of random number generation in Url browsers is discussed and particular threats are analyzed. Through accumulating entropy from the browser, the user interactive operations and local environment variables, we present a secure random number generator completely through ubiquitous Url browser capabilities such as HTTP, JavaScript, AJAX .

We also introduce a new mechanism called Pseudo-cookie for JavaScript programs to access operating system services without changing the Url browser security policies, we exploit the method to retrieve randomness and use it to seed and refresh the state of our generator, which can largely improve the performance of the generator. The security analysis and performance evaluation show promising values for real world applications. As we know, this is the first work addressing the security of random number generation in a pure Url environment.

B.Linear Congruential Generator:

While widely available, the mathematic random function rand()1 in glibc and Math.random() in JavaScript are not feasible for cryptography utilization. The algorithms implemented in Math.random() in Safari 3 and Firefox 2 (we get this information through read the code2,3, we assume other browsers might be the same) are called linear congruential generator, which produce a sequence of numbers x1, x2, . . . according to the linear recurrence xn = axn−1 + b mod m, n ≥ 1; integers a, b and m are parameters of the generator and x0 is the seed. Although this generator provides the uniform distribution random numbers, it does not satisfy the unpredictable requirement. Given a partial output sequence, the remainder of the sequence can be reconstructed even if the parameters a, b and m are unknown.

C.True Random Number Generator :

A TRNG requires a naturally occurring source of randomness such as unpredictable physically procedures. The implementation is through an especial hardware device of software program to collect randomness from precise timing of hardware events to monitoring people behaviors.

Pseudo-random Number Generator:

While true randomness is widely available in the nature, it's hard for deterministic computing system to provide true random number generators through deterministic algorithms. Instead the pseudo-random number generator is used, which extends a short truly random number sequence to a much longer sequence that "appears" to be random. The input to the PRNG is called the seed, while the outputs of the PRNG are called pseudo-random numbers.

D.Random Number Generators in Practice:
For the rarity of true randomness, the output of TRNG is often used as the input of PRNG. Many software random number generators have been proposed, implemented and researched. In a detailed survey of software random number generators are discussed, in a generalized software architecture is introduced, in the model of secure random number generation service is discussed, and are analyses of random number generation on Windows and Linux operating system. As a good engineering practice, pseudo-random number generators have been provided as a system service by modern operating systems. For it is more feasible for OS to collect entropy from hardware events and user inputs. Unix-like operating system implements kernel level pseudo-random number generator and provide the interface through a virtual device /dev/random, while Windows provided similar API to provide random numbers. Different from the random devices above is a device implemented in kernel space. Windows RNG is most implemented in user space, so that the design and implementation of Windows can not resist forward security attack, which is conboundingred a big flaw.

III.THE WORKING PRINCIPLE

We presen,which integrates public key cryptography into url applications without any browser plugins. The public key of is provided by identify basis cryptography, eliminating the need of public key and certificate online retrieval; the private key is supplied by the fragment identifier of the URL.

In this project we have four main modules:
System Initialization
Key Pair Extraction
Encrypt and Signing
Fragment Identifier

System Initialization: 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 identify. 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.

A.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 identify. The PKG will generate a private key according to the identify. 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 identify, the corresponding private key can be extracted from the private matrix SKM and the public key can be extracted from the public matrix PKM.

- Encrypt and Signing: 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.

- Fragment Identifier: In this Module remove the fragment identifier from the URL, to retrieve the page. Domain name ("domain.com") and path inbounding the server ("index. Html") will be sent over the network separately to DNS server and the application server. 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.

B.Requirement Analysis
With the increasing popularity of Url 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 plug-in 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.

C. Purpose:
The purpose of software requirements specification specifies the intentions and intended audience of the SRS.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

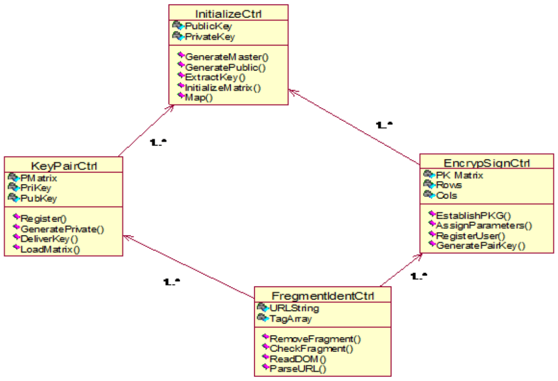
greater harm to enterprise users. Some providers like Google and Yahoo also provide services such as Google Apps for enterprise users to take the place of their own email servers and applications. The misuse of provider's privilege will bring huge losses for their customers.

D.Scope :

The scope of the SRS identifies the software product to be produced, the capabilities, application, relevant objects etc. Public key cryptography basis solutions for the desktop counterpart of the above url applications have been deployed widely for many years. PGP and S/MIME are two de facto standards, and have been implemented within applications inbounding many desktop mail systems. The key management of these solutions requires ad hoc trust management such as PGP "Url 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. 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 url applications

E.Design

ClassDiagram



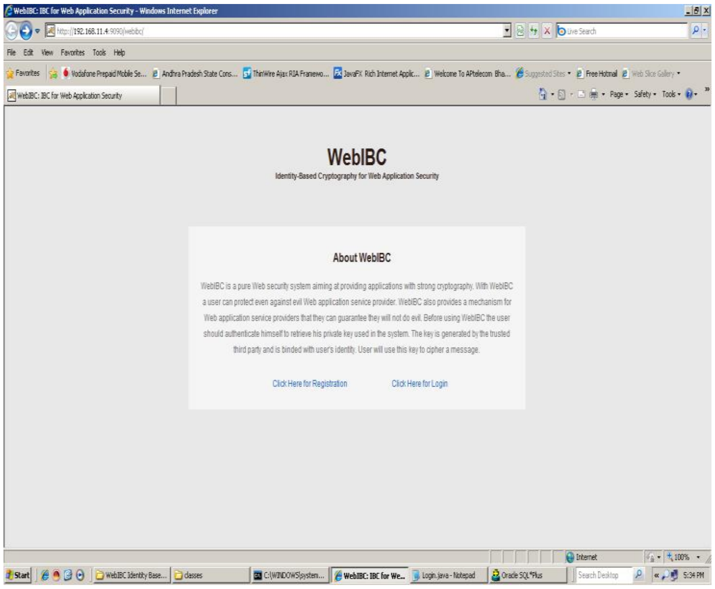
IV.IMPLEMENTATION OF SYSTEM

```
public void calculateLR() throws Exception
        {

                for(int i1=1;i1<=32;i1++)
                        L[0][i1] = IP[i1];
                for(int i2=1;i2<=32;i2++)
                        R[0][i2] = IP[i2+32];


                for(int ilr=1;ilr<=16;ilr++)
                {

                        for(int i1=1;i1<=32;i1++)
L[ilr][i1] = R[ilr-1][i1];


intbr[] =
13,12,13,14,15,16,17,16,17,18,19,20,21,20,21,22,23,24,25,24,
25,26,27,28,29,28,29,30,31,32,1};

int ER[] = new int[49];
for(int i3=1;i3<=48;i3++)
ER[i3] = R[ilr-1][br[i3]];

int KER[] = new int[49];
for(int i4=1;i4<=48;i4++)
{
KER[i4] = k[ilr][i4] ^ ER[i4];
}

int B[][] = new int[9][7];
int j1=1;
int j2=1;
for(int i5=1;i5<=48;i5++)
{
if(j2==7)
{
j1++;
j2=1;
}
B[j1][j2++] = KER[i5];
}

int SBN[] = new int[33];
int isbn=1;

for(int i10=1;i10<=8;i10++)
{
int irow;
int jcol;

irow = 2*B[i10][1] + 1
jcol = 8*B[i10][2] + 4*B[i10][3] + 2*B[i10][4] + 1*B[i10][5];
char finS[] = new char[7];
char tfinS[] = new char[2];
finS[0] = 'S';
//          itoa(i10,tfinS,10);
//          finS[1] = tfinS[0];
finS[1] = (char)i10;
finS[2] = '.';
finS[3] = 't';
finS[4] = 'x';
finS[5] = 't';
finS[6] = '\0';
```

```
String sstr;
sstr = "S";
sstr += i10;
sstr += ".";
sstr += "t";
System.out.println(sstr);

String ssstr = new String(finS);
//          BufferedReader br1= new BufferedReader(new
FileReader(sstr));

int br1[][] = {{0,0,0},

String str1;

int SBox[][] = new int[4][16];
for(int i8=1;i8<=4;i8++)
{
for(int i9=1;i9<=16;i9++)
{
//str1=br1.readLine();
//int temp = Integer.parseInt(str1);
SBox[i8-1][i9-1] = br1[i10][(i8-1)*16+i9];
}
}
public void doPost(HttpServletRequest
request,HttpServletResponse response)
{
int flag = 0;
try
{
PrintWriter out = response.getWriter();
String strmessage = request.getParameter("message");
System.out.println(request.getRequestURI());
DES d2 = new DES();
d2.initialize();
String plain = d2.decrypt(strmessage);
System.out.println(plain);
out.println("Decrypted Message Received: " + plain);
}
catch(Exception ex)
{
ex.printStackTrace();
}int Svalue;
Svalue = SBox[irow][jcol];
int B4[] = new int[5];
B4[4] = Svalue % 2;
Svalue = Svalue / 2;
B4[3] = Svalue % 2;
Svalue = Svalue / 2;
B4[2] = Svalue % 2;
Svalue = Svalue / 2;
B4[1] = Svalue % 2;
Svalue = Svalue / 2;

SBN[isbn++] = B4[1];

}
```

V.EXPERIMENTAL RESULTS

The concept of this paper is implemented and different results are shown below.



WebIBC

Identity-Based Cryptography for Web Application Security

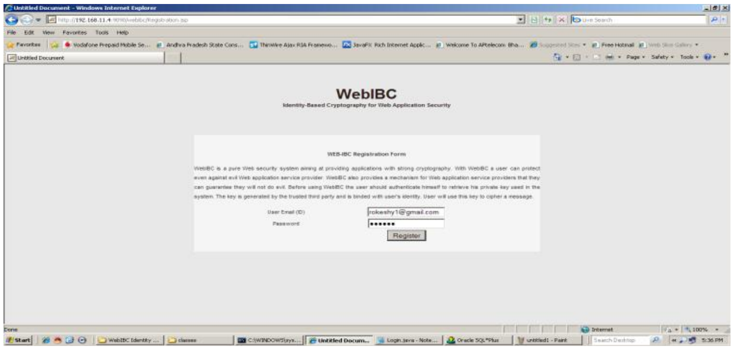
About WebIBC

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

Click Here for Registration

Click Here for Login

MAIN PAGE



WebIBC

Identity-Based Cryptography for Web Application Security

WEB-IBC Registration Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

User Email (ID): rokeshy1@gmail.com

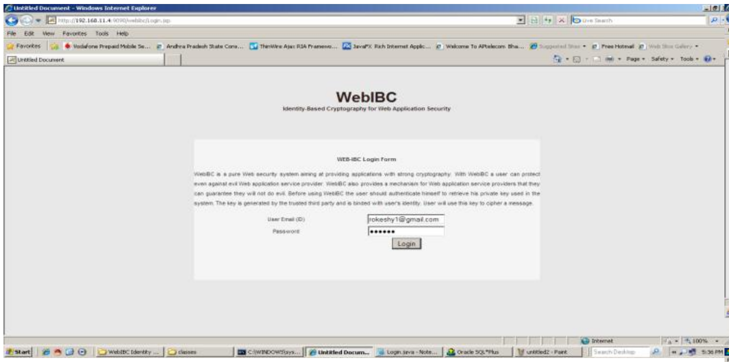
Password: ••••••

Register

REGISTRATION FORM

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering,
Vol. 1, Issue 7, September 2012



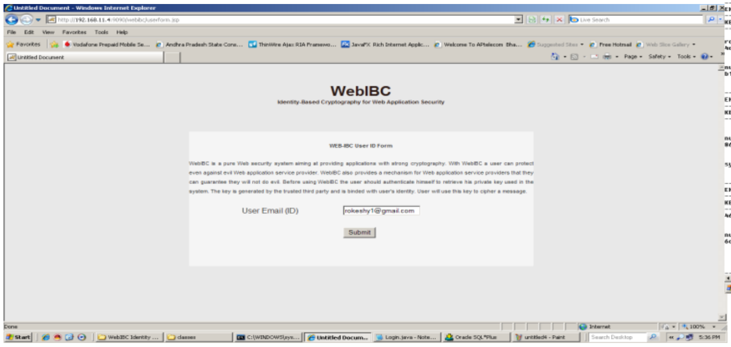
WebIBC
Identity-Based Cryptography for Web Application Security

WEB-IBC Login Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

User Email (ID): rokeshy1@gmail.com
Password: ••••••

WERIBC LOGIN FORM



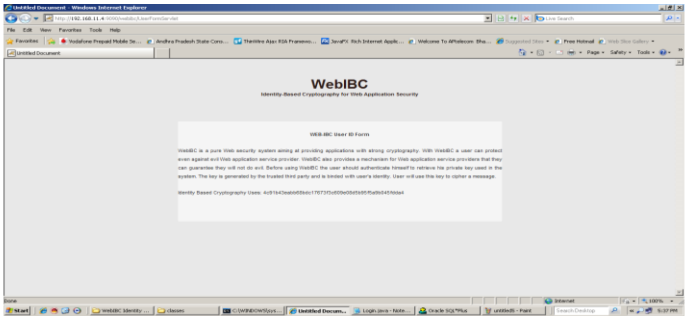
WebIBC
Identity-Based Cryptography for Web Application Security

WEB-IBC User ID Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

User Email (ID): rokeshy1@gmail.com

USER ID FORM



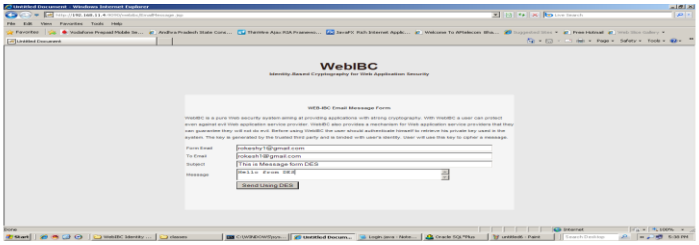
WebIBC
Identity-Based Cryptography for Web Application Security

WEB-IBC User ID Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

Identity Based Cryptography Uses: 4c91b43eabb68bdc17673f3c609e08d5b95f5a9b045fdda4

GENERATED KEY BASIS ON USER ID



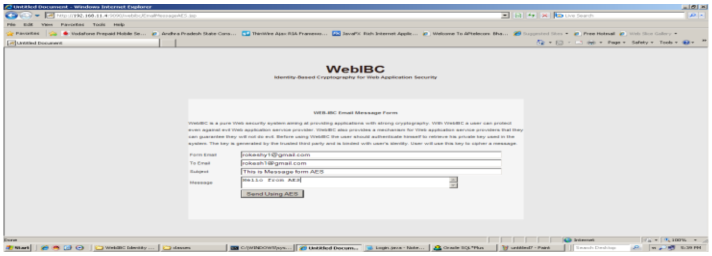
WebIBC
Identity-Based Cryptography for Web Application Security

WEB-IBC Email Message Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

Form Email: rokeshy1@gmail.com
To Email: rokesh1@gmail.com
Subject: This is Message form DES
Message: Hello from DES

E-MAIL MESSAGE FORM FOR DES



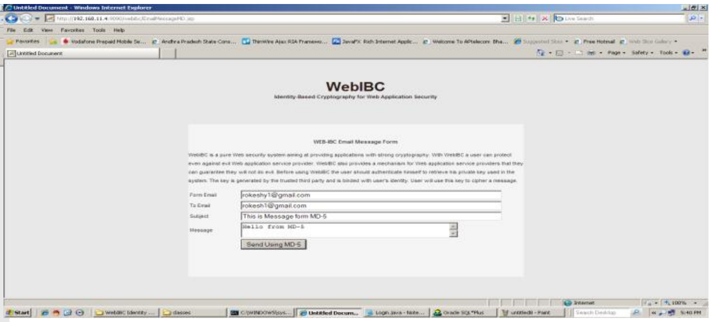
WebIBC
Identity-Based Cryptography for Web Application Security

WEB-IBC Email Message Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

Form Email: rokeshy1@gmail.com
To Email: rokesh1@gmail.com
Subject: This is Message form AES
Message: Hello from AES

E-MAIL MESSAGE FORM FOR AES



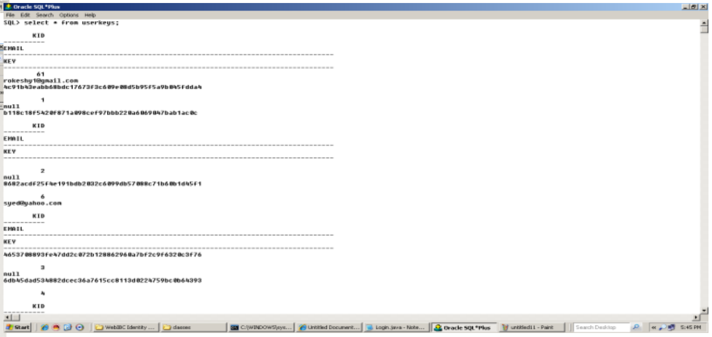
WebIBC
Identity-Based Cryptography for Web Application Security

WEB-IBC Email Message Form

WebIBC is a pure Web security system aiming at providing applications with strong cryptography. With WebIBC a user can protect even against evil Web application service provider. WebIBC also provides a mechanism for Web application service providers that they can guarantee they will not do evil. Before using WebIBC the user should authenticate himself to retrieve his private key used in the system. The key is generated by the trusted third party and is binded with user's identity. User will use this key to cipher a message.

Form Email: rokeshy1@gmail.com
To Email: rokesh1@gmail.com
Subject: This is Message form MD-5
Message: Hello from MD-5

E-MAIL MESSAGE FORM FOR MD-5



```
SQL> select * from userkeys;

       KID
----------
EMAIL
--------------------------------------------------------------------------------
KEY
--------------------------------------------------------------------------------
        61
rokeshy1@gmail.com
4c91b43eabb68bdc17673f3c609e08d5b95f5a9b045fdda4

         1
null
b118c18f5420f871a098cef97bbb220a6069047bab1ac0c

       KID
----------
EMAIL
--------------------------------------------------------------------------------
KEY
--------------------------------------------------------------------------------

         2
null
8682acdf25f4e191bdb2032c6099db57088c71b60b1d45f1

         6
syed@yahoo.com

       KID
----------
EMAIL
--------------------------------------------------------------------------------
KEY
--------------------------------------------------------------------------------
46537088893fe47dd2c072b128862960a7bf2c9f6320c3f76

         3
null
6db45dad534882dcec36a7615cc8113d0224759bc0b64393

         4

       KID
----------
```

STORED USER ID KEYS

VI.CONCLUSION

In this project, we present to protect the system bounding security and privacy of url applications. integrates identify basis cryptosystem into url basis applications and is totally established by java without any browser plugins. We have implemented a prototype of and performance evaluation indicates its effectiveness and efficiency over other systems. The security analysis shows that is resilient to some known attacks using the proposed schemes. The future work of is to evaluate the feasibility of other schemes on.

REFERENCES

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

[2] C. Jackson, A. Barth, A. Bortz, W. Shao, and D. Boneh. Protecting browsers from dns rebinding attack. ACM Conference on Computer and Communications Security, 2007.

[3] K. Paterson. ID-basis signatures from pairings on elliptic curves, cryptology eprint archive, report 2002/004. http://citeseer.ist.psu.edu/paterson02idbasis.html.

[4] A. Shamir. Identify-basis cryptosystems and signature schemes. Crypto '84, pages 47– 53, 1985.

[5] Pseudo-randomness Inbounding Url Browsers, Zhi Guan, Long Zhang, Zhong Chen, and Xianghao Nan, Springer-Verlag Berlin Heidelberg 2008

[6] An Economical Model for the Risk Evaluation of DoS Vulnerabilities in Cryptography Protocols, Zhen Cao, Zhi Guan, Zhong Chen, Jianbin Hu, and Liyong Tang, Springer-Verlag Berlin Heidelberg 2007

[7] C. Cocks. An identify basis encryption scheme basis on quadratic residues. Lecture Notes in Computer Science, 2260:360–363, 2001.

[8] D. Hankerson, A. Menezes, and S. Vanstone. Guide to elliptic curve cryptography. Springer-Verlag, 2004.

[9] The description of Software Requirements Specifications is derived from IEEE std. 830-1993.The Unified Modeling Language by Grady Booch.

[10]. Zhi Guan, Zhen Cao, Xuan Zhao, Ruichuan Chen, Zhong Chen, Xianghao Nan” : Identify Basis Cryptography for System Bounding Security in Url Applications”.

Biography

Mr.K.Ramakrishna Graduted In Information Technology And Engineering Form Kakatiya University ,Warangal.And M.Tech In Software Engineering From JNTU Hyderabad, India.He Is Working Presently As Assistant.Professor In Department Of Computerscience Engineering In Holy Mary Institute Of Technology And Science,Hyderabad,India,He Is Has 4+ Years Experience,His Research Interests Include Mobile Communication And Networking.

Mr B.Srinivasulu, Post Graduated in CSE(M.Tech) From JNTUH, 2010, and graduated in Computer Science & Engineering (B.TECH) From JNTU Hyderabad, 2008. He is working presently as Asst.Professor in Department of Computer Science & Engineering in HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE (HITS), R.R.Dist, A.P, INDIA.He Is Has 2+ Years Experience,His Research Interests Include Data Warehousing & Data Mining and Cloud Computing.

Rakesh.U, B.Tech(CSIT) in SKTRM college of engineering,JNTUH, and is M.Tech In Software Engineering from Jawaharalal Nehru Technological University Hyderabad,A.P,India in 2010.He Is Working presently as Assistant Professor In Department of Computer Science And EngineeringInMurthyInstituteOfTechnologyand Science,R.R.District ,A.P,INDIA.He has 2 Years experience.

Mr.B.SatheeshKumar, Graduated in B.Tech(CSIT) SKTRM college of engg From JNTU, Hyderabad, Andhra Pradesh, India.And Is M.Tech In CSE From JNTUH,Hyderabad,A.P,India.He Is Working Presently As Assistant Professorin department Of Computer ScienceAndEngineering in Bandaru Srinivas Insti Oftechnology And Science,He Is Having 2 Years Exp. His Research Interests Include Mobile Communication.

Mr M.Srinivasa Rao Post Graduated in Computer Applications (MCA) From Acharya Nagarjuna University, 2005 and post graduated in Computer Science & Engineering (M.TECH) From JNTU Hyderabad, 2012. He is working presently as Asst.Professor in Department of Computer Science & Engineering in HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE (HITS), R.R.Dist, A.P, INDIA.