International Journal of Security and Its Applications
Vol. 11, No. 8 (2017), pp.13-26
http://dx.doi.org/10.14257/ijsia.2017.11.8.02

Extended Design and Implementation of Certificate Authorities

SarveshTanwar* and Anil Kumar

Dept. of Computer Science & Engineering
Mody University of Science & Technology, India
s.tanwar1521@gmail.com

Abstract

The most important security services of Public Key Infrastructure (PKI) such as e authentication, integrity, confidentiality and non- repudiation enables its clients to maintain a level of trust. It enables clients to exchange information over unsecure public network such as Internet. PKI proves the identity of an individual or an organization via digital certificates which binds information of client and public key. Public keys are store in public key directory. A PKI system works by having a Certificate Authority (CA) that is responsible for issuing and revoking certificates. Certificates are basic source of trust in online transactions. The aim of this paper is to design and implement a CA that can create and manage public key certificates. We have proposed a trusted hierarchical trust model which is extension of work done by Janabi et al.. [4]. The proposed system is designed and implemented using JAVA programming language, MYSQL database server and Apache web server.

***Keywords**: Public Key Infrastructure, Certificate Authority, Public Key Certificate, Hierarchical Trust Model.*

1. Introduction

All information sent to the Internet is basically public and accessible by all; the need for security becomes critical. The most critical element of security is the ability to provide trust among users and confidence for reliable transactions over the insecure Internet.
Today PKI can be viewed as critical for the commercial sectors and the government. The Public-key system makes it possible for two parties belonging to different domains to communicate securely without either having to know or trust the other party [4]. Both the parties trust on the trusted third party, known as CA who identifies and certifies that their keys are genuine.

CA is the root of the hierarchical tree. CA guarantees that they are who they claim to be [12]. CA is responsible for issuing and maintaining certificates [10]. Various institutions and organizations can utilize this security technology to satisfy current business needs.

1.1 Public Key Infrastructure (PKI)

A PKI is the combination of well defines standards, software, hardware, key generation, encryption technologies, certificate generation and distribution processes and services. It enables secure and safe online transactions by the exchange of digital certificates between users [11][13] .

A PKI (Trust Hierarchy*)* consists of:

a) **Certification Authority (CA):** It is the most trusted authority in a network that issues and manages security credentials and public keys for certificate and signs the

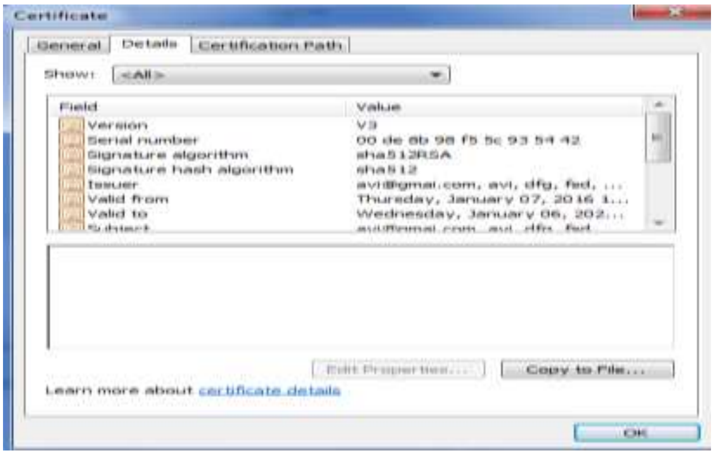
ISSN: 1738-9976 IJSIA

certificates. Registration Authority (RA) or Sub CA verifies requestor's information and CA/Sub CA can issue a certificate. In fact, the CA is responsible for the distribution and revocation of the certificate.

b) **Sub Certificate Authorities (Sub CA):** Sub Certificate Authorities are sometimes called registration authorities. A sub CA performs administrative tasks of a CA such as to verify the identity of the end user and determine if an end entity is entitled to have a public key certificate issued [4].

c) **The End-users:** End users are the key element of the PKI who are using the application. They request for a certificate which is issued by CA/Sub CA.

d) **Public Key Certificates (PKC)-** Just like a license or a passport, a digital certificate is a form of identification. It provides information about the identity of user like name, address, contact number, organization, valid from, valid to and subject's public key as well as other supporting information. It is issued and verified by a CA who guarantees the validity and authenticity of the information in the certificate. There are different types of certificates such as PGP (Pretty Good Privacy), SPKC (Simple Public Key Certificate) and X.509. X.509 certificates are used by most of the organizations [13]. Figure 1 illustrates structure of a certificate.



Show: <All>

Field	Value
Version	V3
Serial number	00 de 8b 98 f5 5c 93 54 42
Signature algorithm	sha512RSA
Signature hash algorithm	sha512
Issuer	avi@gmail.com, avi, dfg, fed, ...
Valid from	Thursday, January 07, 2016 1...
Valid to	Wednesday, January 06, 202...
Subject	avi@gmail.com, avi, dfg, fed

Figure 1. Structure of X.509 version 3 Certificate

e) **Certificate Policy (CP):**A document that sets out the rights, duties and obligation of each party in a PKI. The purpose of this document is to establish a relationship between certificate policies and CPSs. CP is a set of rules and commitments made by a CA to indicate the applicability of a certificate to a particular set of applications or group of users. The main objective of CP is to determine the security policy that is followed by a certification organization. Also, it can be used as a reference to establish a domain-trust relation with this organization in a transaction with other organizations.

f) **Directory:** Certificate directory or repository store all the certificates and user profile information in a database. Certificate Revocation Lists (CRL) can also be stored in the repository.

g) **Archive:** Archive for the safe and long-term storage of information [4]

1.2 Trust Models

A trust model is a collection of rules that inform application how to decide the legitimacy of a Digital Certificate [16]. Most widely used PKI trust models are: Peer to Peer, Hierarchical, Bridge, Mesh and Hybrid [9][11].

1.2.1 Single CA PKI Architecture

Single CA PKI architecture is one which contains one CA who issues and distributes certificate and CRLs to the entities or users. All these users trust this CA. The single CA architecture does not allow for any new CA to be added to the PKI. As there is a common point of trust, all the entities can communicate with each other in a trusted environment. But it also presents a single point of failure because there is only one CA that holds the key information of all the users.

If the private key of the CA has been compromised, this might result in a complete breakdown of the PKI system. All the users in this architecture communicate with each other in a trusted environment, as there is a common point of trust, which is CA1. For example user1 and users2 are two entities who trust CA, CA1. Therefore both of them can validate and verify each other's certificates and can communicate. The certificate path of user1 can be depicted as follows:- CA1-> user1

1.2.2 Peer to Peer Trust Model

In Peer to Peer architecture, there is no starting point as a trusted root CA. Certificate users typically rely on their own local CA, and as a starting point of trust. The two CA are isolated. They are different trust domains; domain users can verify the domain user. This trust model is the most prominent feature is its flexibility, making the trust domain extension is very convenient. But it is this flexibility, so that the manageability of the whole system worse off. The certificate path building is a very difficult task when increasing number of CA. There may be multiple certification paths, or may be an infinite loop, thus making it difficult to verify the certificate, thereby increasing the burden on users. Therefore, such model is applicable to small number of groups of coequal status of the implementation of the organizational PKI.

1.2.3 Bridge CA

The Bridge CA acts as a mediator between different organizations which interconnects different PKIs to establish trust path among them. Two organizations have a trusted path through Bridge CA, where the certificate policies overlap.

BCAs can decrease the total number of cross certificates required to join the PKIs. The BCA does not become a trust anchor for any of the PKIs as it is not directly trusted by any of the PKI entities. Rather trust is referenced from internal PKI trust anchors. The United States federal PKI (FPKI) project is attempting to join together multiple PKIs set up under separate federal agency programs using bridge CAs [14][11].

1.2.4 Hierarchical Trust Model

Hierarchical architecture is like an inverted tree structure, in which root is the starting point of trust. The root CA (RCA) issues certificates to subordinate CAs, whereas subordinate CAs may issue certificate to other CAs or users. The trust relationship is specified in only one direction from RCA to users. All the users trust on RCA. If crises occur on RCA that break down confidence throughout the PKI system. In this architecture certificate path begins with the root CA's public key.

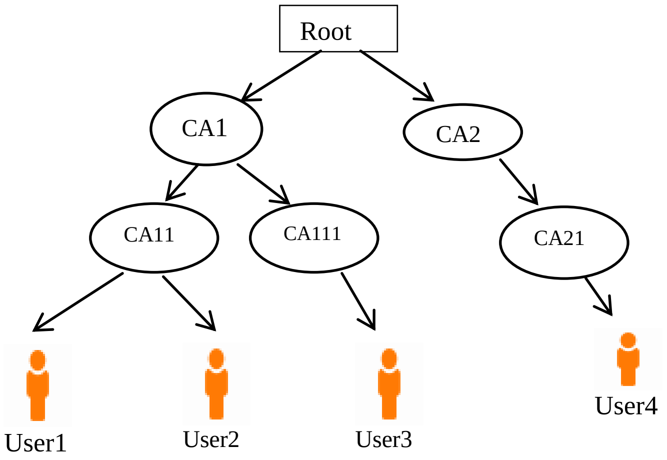


Figure 2. Hierarchical Trust Model

In this example, let's suppose that the subject "user 2" needs to certify a message from the subject "user 4". It will need to go through the certification path composed by CA11 -> CA1 ->RootCA-> CA2 -> CA21. But if several subjects from CA11 must certify messages coming from subjects from CA21, a cross-certificate may be used and then path can be reduced to CA11-> CA21

The paper is organized in the following sections - section 2 describe related works, Design and Implementation of algorithm is given in section 3 along with results followed by security analysis in section 4 and finally section 5 is the conclusion of the paper.

2. Related Work

Security plays an important role in the communication systems. In the following section we will review the PKI based papers.

Gaya (2000) [15] explained need for security, PKI framework, PKI enabled application and PKI issues such as – private key management, CA cross certificates, deployment and interoperability.

Subramanya *et al.* (2006) [8] described digital signature generation and verification process based on RSA and DSS. They said in traditional and newer business application have carried out electronic transactions, which lead to critical need for security of information from alteration, ensures authenticity, integrity and non-repudiation. Digital signature serves the purpose of authentication and validation of electronic transactions.

Liping *et al.* (2011) [6] Describes the related concepts. They command do the theoretical comparison of different PKI-based trust models- Hierarchical trust model, Peer to Peer trust model, Network trust model and hybrid trust model.

Kharche *et al.* (2012)[3] proposed a solution to build trust in cloud using PKI. The proposed scheme consists of enterprise Root CA, enterprise CA and End Entities (EE). EE should trust on enterprise RCA and enterprise CA. The drawback of the proposed approach is that certificate issued by RCA will be supported by few browsers.

Janabi *et al.* (2012) [4] said that the most important security services are confidentiality, integrity, authentication, and non-repudiation. When designing a communication system, the security services must be considered. In PKI system, CA issue public-key certificates. The author aims to design and implement a CA to create and assign public key certificates for web application. Hence, the system enables secure communication and proper authentication to facilitate the revocation of the certificates. PHP and HTML programming languages besides Apache web server and MyfSQL database server were used to designed and implemented the proposed system.

Jøsang *et al.* (2013) [2] described PKI as set of policies, procedures and technologies for building trust among end entities where authentication is needed in online transactions.

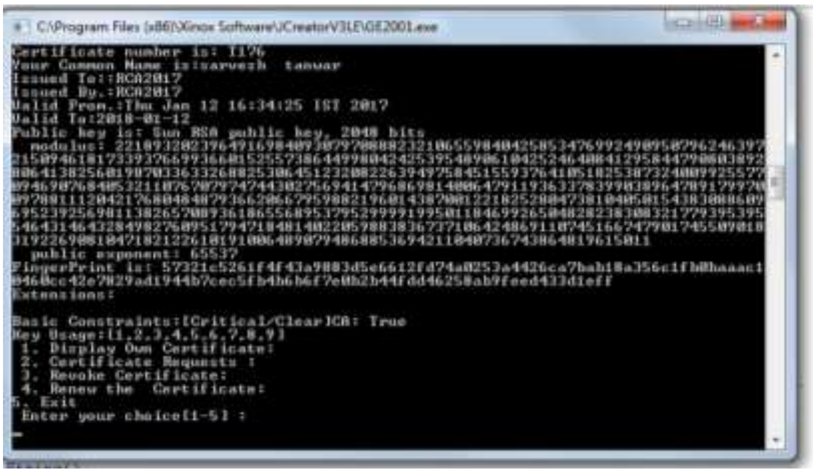
Nandhini M. *et al.* (2015) [1] described PKI as a solution to satisfy the requirements in availability, privacy and scalability. They introduce some novel mechanisms to satisfy the security requirements. They proposed a solution for the Denial of Service attacks in PKI for different application categories. They simulate the proposed mechanism on NS2 simulator. Their main objective is to give basic idea of how the simulator work, how to setup simulation networks and how to create new network components.

A lots of work has been done on PKI. In Janabi [4] author has proposed PKI for web application. Author has developed the system using php and MySql for database. According to author this work can be extended to hierarchical trust model and author has not provided security to the private key. Hence this paper is extended work of Janabi [4] work.

3. Design And Implementation of Certificate Authorities

The system proposed by Janabi [4] considers only one domain of trust. The author said it is important to extend the system to hierarchical domain trust. We have proposed hierarchical trust model with private key security as well as data base security.

Step 1: CA is given a valid user id and password. Using it can sign in and fills his profile and generates a digital certificate. In response a key pair is generated. CA then encrypts the certificate using his private key.



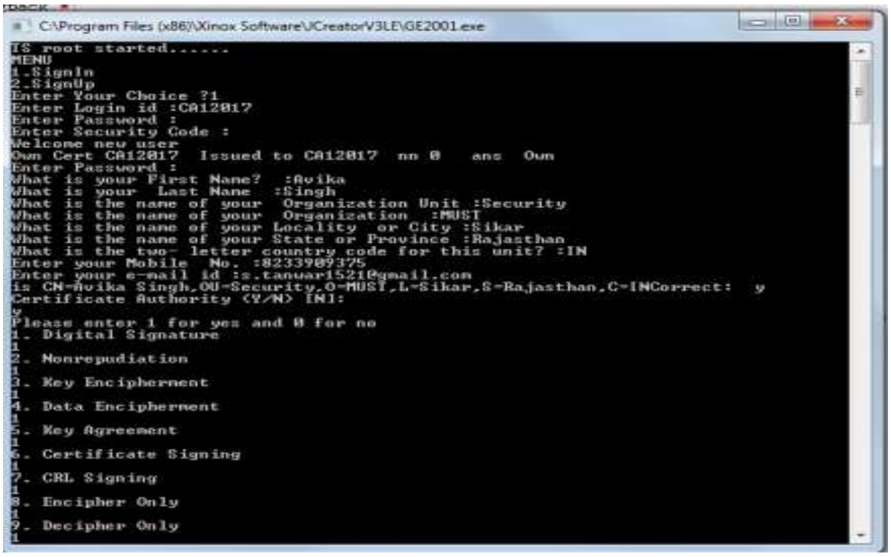
C:\Program Files (x86)\Xinox Software\JCreatorV3LE\GE2001.exe

```
Certificate number is: 1176
Your Common Name is:sarvesh  tanwar
Issued To::RCA2017
Issued By.:RCA2017
Valid From.:Thu Jan 12 16:34:25 IST 2017
Valid To:2018-01-12
Public key is: Sun RSA public key, 2048 bits
  modulus: 22109320239649169840930797008823210655984042585347699249095087962463977
21509461817339376699366015255573864499800424253954090610425246400841295844790603892
80641382560190703363326882530645123208226394975845155937641051825387324009925577
09469076840532110767079747443027569414796869814006479111936337839903896470917997
09700111204217680048407936620667959082196014387001221025200473010400501543830000609
695239256901138265700936186555689537952999991995011846992650482823830032177939539
546431464328490276095179471848140220598838367371064248691107451667479017455090100
31922690010471821226101910064090794068053694211040736743064019615011
  public exponent: 65537
FingerPrint is: 57321c5261f4f43a9803d5e6612fd74a0253a4426ca7bab18a356c1fb0baaac1
0460cc42e7029ad1944b7cec5fb4b6b6f7e0b2b44fdd46258ab9feed433d1eff
Extensions:

Basic Constraints:[Critical/Clear]CA: True
Key Usage:[1,2,3,4,5,6,7,8,9]
 1. Display Own Certificate:
 2. Certificate Requests :
 3. Revoke Certificate:
 4. Renew the  Certificate:
5. Exit
 Enter your choice[1-5] :
_
```

Figure 3. Root CA's Certificate

Step 2: When a new sub CA applies for certificate request, he/she fills a sign up form using createadmin class.


C:\Program Files (x86)\Xinox Software\JCreatorV3LE\GE2001.exe

```
IS root started......
MENU
1.SignIn
2.SignUp
Enter Your Choice ?1
Enter Login id :CA12017
Enter Password :
Enter Security Code :
Welcome new user
Own Cert CA12017  Issued to CA12017  nn 0   ans  Own
Enter Password :
What is your First Name?  :Avika
What is your  Last Name  :Singh
What is the name of your  Organization Unit :Security
What is the name of your  Organization  :MUST
What is the name of your Locality  or City :Sikar
What is the name of your State or Province :Rajasthan
What is the two- letter country code for this unit? :IN
Enter your Mobile  No. :8233909375
Enter your e-mail id :s.tanwar1521@gmail.com
is CN=Avika Singh,OU=Security,O=MUST,L=Sikar,S=Rajasthan,C=INCorrect:   y
Certificate Authority (Y/N) [N]:
y
Please enter 1 for yes and 0 for no
1. Digital Signature
1
2. Nonrepudiation
1
3. Key Encipherment
1
4. Data Encipherment
1
5. Key Agreement
1
6. Certificate Signing
1
7. CRL Signing
1
8. Encipher Only
1
9. Decipher Only
1
```

Figure 4. Certificate Request by CA/Sub CA

On acceptance of account creation request a certificate generation form is filled by sub CA. A key pair is generated for sub CA. Then his digital certificate is signed by CA's private key.

Step 3: First of all a new user fills up sign up request having various fields such as Common Name (CN), Organization Name (O), Organization Unit (OU), City (CT), State (S) , Country (C), Email (E) and Mobile Number (MN) using create admin class and store all the values in the database.

$$\begin{aligned}
&String\ concat1\\
&\qquad = srno + oo.getFname() + oo.getLname() + oo.getCity()\\
&\qquad + oo.getState() + oo.getOrgunit() + oo.getOrg()\\
&\qquad + oo.getEmail() + oo.getCountrycode() + oo.getMobileno()\\
&\qquad + oo.getIssuedBy() + oo.getIssuedTo() + oo.getValidto()\\
&\qquad + oo.getValidfrom() + publickey.toString();\\
&System.out.println(\text{"}Fingerprint = \text{" } + msgdigest);
\end{aligned}$$

$$\begin{aligned}
&msgdigest = new\ sha512().sha512(concat1);\\
&oo.setDigest(msgdigest);\\
&byte[]\ b = new\ byte[1024];\\
&\quad b = concat1.getBytes();\\
&msgdigest = sh.new\ Digest().digestIt(b);
\end{aligned}$$

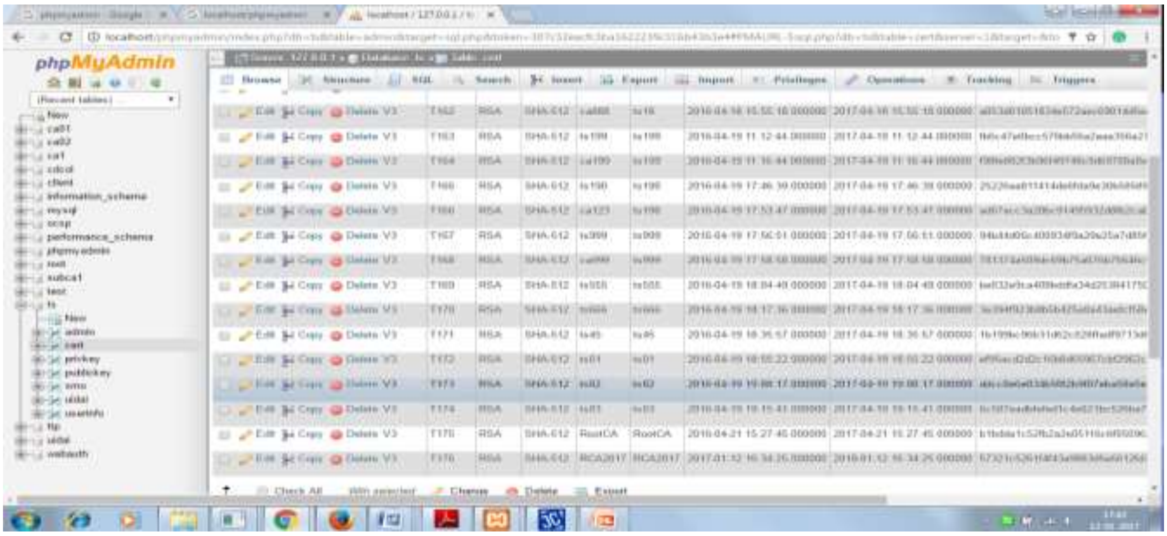
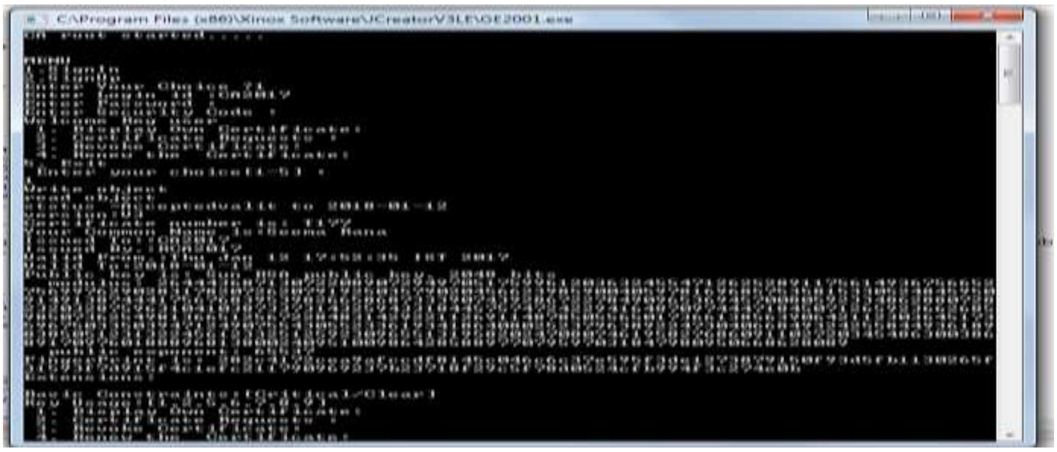


Figure 5. Root CA Database

Step 4: When CA accepts the account creation request, a form is automatically sent to the user regarding digital certificate creation.



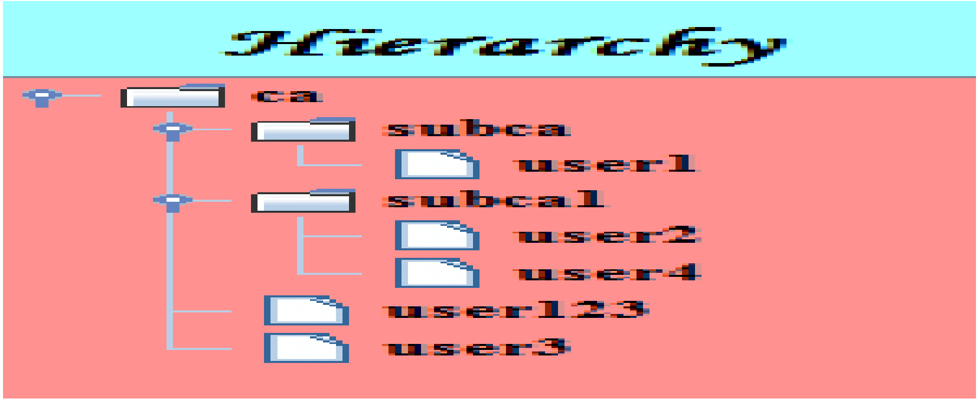
```
C:\Program Files (x86)\Xinox Software\JCreatorV3LE\GE2001.exe
CA root started.....
[illegible]
```

Figure 6. Sub CA Certificate

The user now sends the certificate request to sub CA which then validates the data and forwards the request to CA.

Step 5: Either the user or CA on behalf of user can generate key pair (Private-Public key) and store keys in blob form in the database, which are accessed directly in the form of object from the database to sign the digital certificate. The hierarchical tree is updated accordingly. Figure 7 shows hierarchy of CAs.

```
bos = new ByteArrayOutputStream();
   oos = new ObjectOutputStream(bos);
oos.writeObject(obj);
data = bos.toByteArray();
Class.forName("com.mysql.jdbc.Driver");
 System.out.println("Connecting to database...");
```



Hierarchy

- ca
  - subca
    - user1
  - subca1
    - user2
    - user4
  - user123
  - user3

Figure 7. Overall Hierarchy of the Organization

Step 6: The user can log in into his account using his unique Id, password and security code. On log in it can click on "view self-certificate".

Step 7: The certificate of Sub CA is decrypted by CA's public key and user's certificate is decrypted using public key of Sub CA, if certificate is used by Sub CA.

Step 8: Now CA/Sub CA/User and view certificate which binds his public key. Figure 3,4 and 6 shows the certificate output and certificate path.

Step 9: After certificate Generation and Creation next step is to transit the certificate to the recipient. Whether the certificate is received on time or not there will be handshaking process using shared key (AES -128 bit). Certificate is encrypted using shared key and share key is encrypted by public key of the receiver alongwith a timestamp so that only intended recipient can open it.

$byte\ []\ bbskey\ =\ certWrt.getSemkey();$

$E_{sh}(Cert_{user}\ ,time\ stamp)$
$E_{pub_u}(sh_u)$

Step 10: User extract encrypted symmetric key and decrypt it by its own private key and

$$byte[]\ bb\ =\ se.decryptData(bbskey, ownprikey, "RSA/ECB\ /PKCS1Padding");$$

3.1 Flow chart of working:

The following flow chart depicts the working of proposed scheme.

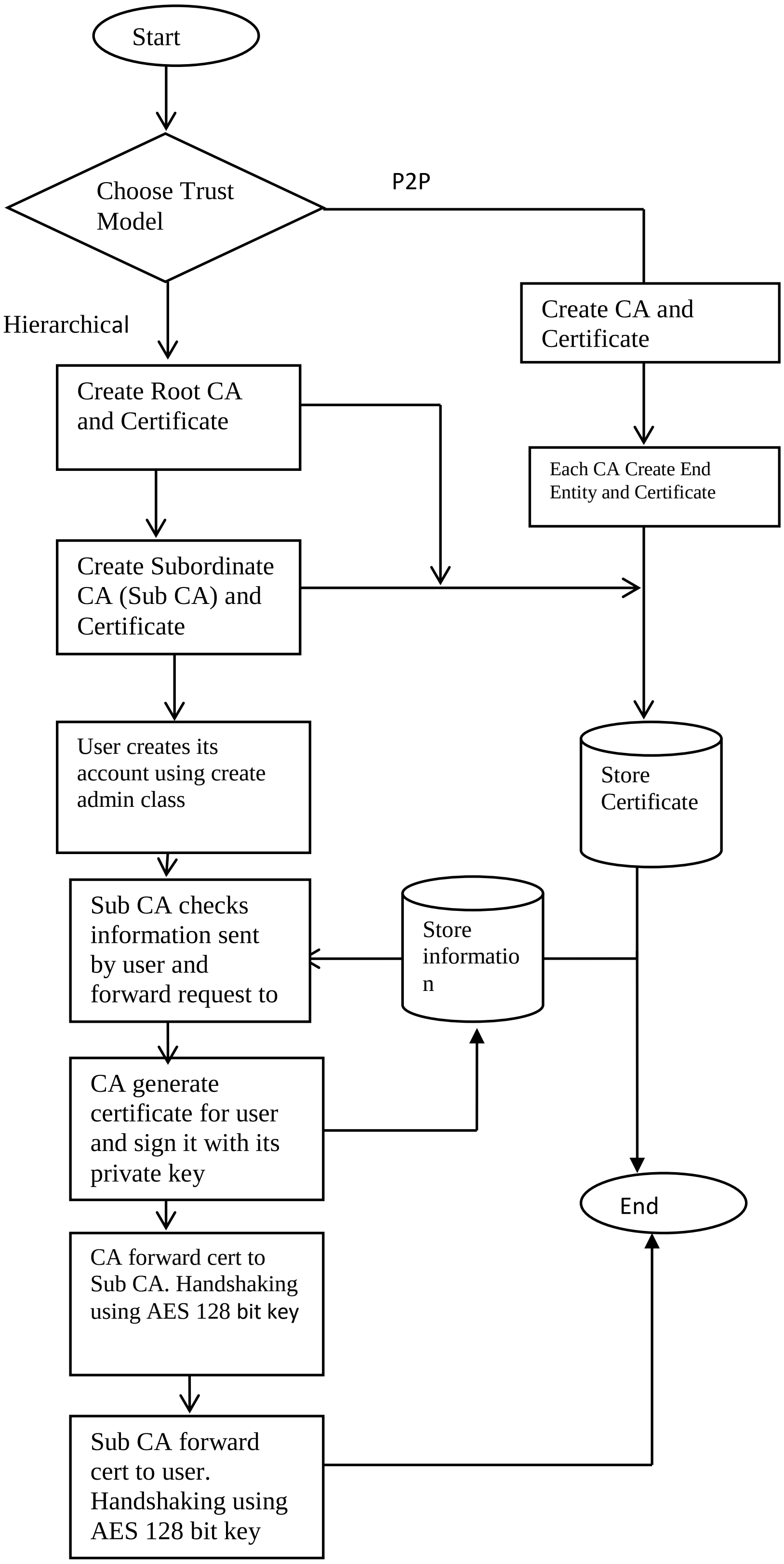
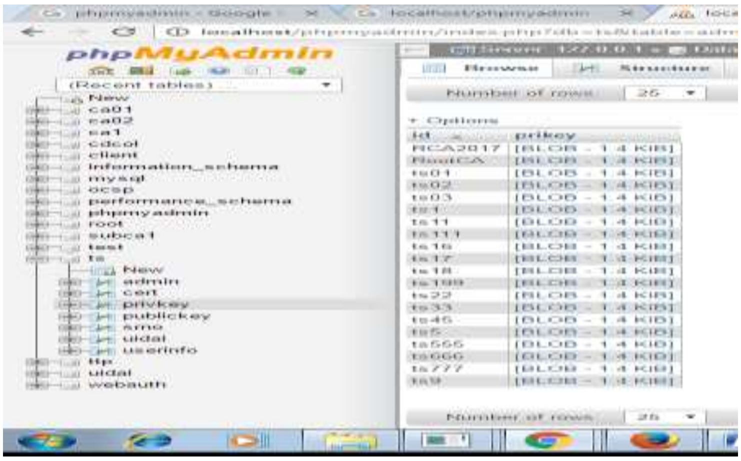

Start
Choose Trust Model
P2P
Hierarchical
Create CA and Certificate
Create Root CA and Certificate
Each CA Create End Entity and Certificate
Create Subordinate CA (Sub CA) and Certificate
User creates its account using create admin class
Store Certificate
Sub CA checks information sent by user and forward request to
Store informatio n
CA generate certificate for user and sign it with its private key
End
CA forward cert to Sub CA. Handshaking using AES 128 bit key
Sub CA forward cert to user. Handshaking using AES 128 bit key

Figure 8. Flow Chart of the Working of Implementation of Hierarchal Model in PKI

4. Security Analysis

The designed trust model is secure as compared to the existing one as it uses hybrid cryptography with Message Digest of SHA-512. When RCA/CA/SubCA and User perform SignUp it will ask about security code and password. When user type password that are not visible on the screen e.g is hidden so that no one can find out password and security code.

1. ***Trust prints:*** *All the end entities can trust on RCA and CA as they are inside their boundaries.*
2. ***Security properties of PKI:*** *This approach gets the full functionalities of PKI, key management and trust between entities.*
3. **Private Key management:** Private Key storage is one of the crucial issues in PKI deployment. Private Key can be stored either on local machine, disks or smart cards. To store private key on these storage is insecure [17] as unauthorized users can access machine or disks and smart cards may be lost. These storage devices are weak in security. In proposed approach private key is stored in blob form. Server fires a request to the database server. Result obtained is thereafter sent to the end user via Server. In our model it becomes very easy for the server who is client for the database server to securely store, easily manage, efficiently access and successfully retrieve bulky data with quick response. The data is stored as attributes of an Object. This object of a class is the converted into Byte Array. This Byte Array is stored in BLOB data type of the database. The private key is stored in an object of class RSA which is present in BLOB form in database. So first the BLOB is retrieved in byte array stream, converted into RSA class object and now the private key is retrieved.

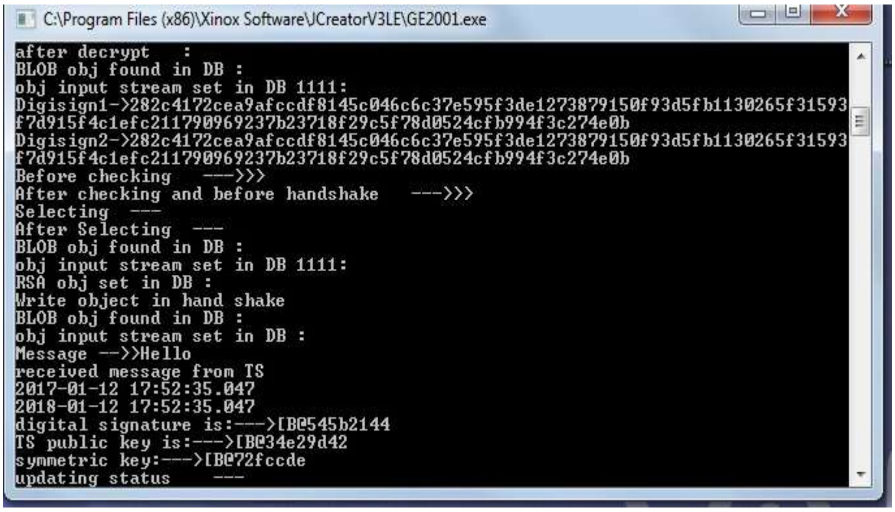


id	prikey
RCA2017	[BLOB - 1.4 KiB]
RootCA	[BLOB - 1.4 KiB]
ts01	[BLOB - 1.4 KiB]
ts02	[BLOB - 1.4 KiB]
ts03	[BLOB - 1.4 KiB]
ts1	[BLOB - 1.4 KiB]
ts11	[BLOB - 1.4 KiB]
ts111	[BLOB - 1.4 KiB]
ts16	[BLOB - 1.4 KiB]
ts17	[BLOB - 1.4 KiB]
ts18	[BLOB - 1.4 KiB]
ts199	[BLOB - 1.4 KiB]
ts22	[BLOB - 1.4 KiB]
ts33	[BLOB - 1.4 KiB]
ts45	[BLOB - 1.4 KiB]
ts5	[BLOB - 1.4 KiB]
ts555	[BLOB - 1.4 KiB]
ts666	[BLOB - 1.4 KiB]
ts777	[BLOB - 1.4 KiB]
ts9	[BLOB - 1.4 KiB]

Figure 9: Private Key storage in Blob Form

4. ***Handshaking with time Stamping***

As certificate received by the CA/Sub-CA/User acknowledgement will be sent to RCA/CA/Sub-CA respectively. For this purpose a handshaking process encrypted with shared key and a timestamp required for the validity of the certificate.



C:\Program Files (x86)\Xinox Software\JCreatorV3LE\GE2001.exe

```
after decrypt   :
BLOB obj found in DB :
obj input stream set in DB 1111:
Digisign1->282c4172cea9afccdf8145c046c6c37e595f3de1273879150f93d5fb1130265f31593
f7d915f4c1efc211790969237b23718f29c5f78d0524cfb994f3c274e0b
Digisign2->282c4172cea9afccdf8145c046c6c37e595f3de1273879150f93d5fb1130265f31593
f7d915f4c1efc211790969237b23718f29c5f78d0524cfb994f3c274e0b
Before checking   --->>>
After checking and before handshake   --->>>
Selecting  ---
After Selecting  ---
BLOB obj found in DB :
obj input stream set in DB 1111:
RSA obj set in DB :
Write object in hand shake
BLOB obj found in DB :
obj input stream set in DB :
Message -->>Hello
received message from TS
2017-01-12 17:52:35.047
2018-01-12 17:52:35.047
digital signature is:--->[B@545b2144
TS public key is:--->[B@34e29d42
symmetric key:--->[B@72fccde
updating status    ---
```

Figure 10: Handshaking and timestamp between RCA and CA

5. **Public key directory**
   This will also maintain a public key directory where all the public keys are stored in the form of object. This will prevent public key replacement attack because public key can't be directly accessed.

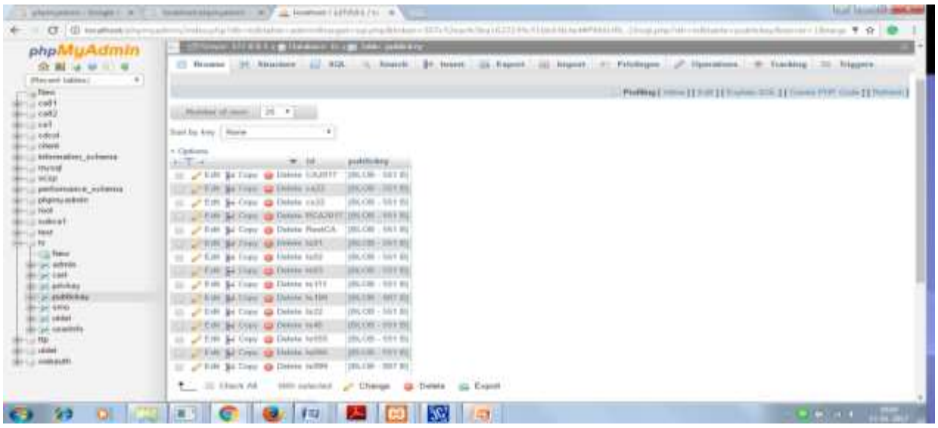


Figure 11: Public key directory

6. **Peer authentication:-** The peer's identity can be authenticated using asymmetric key cryptography (RSA 2048 bits/4096bits)

7. **Reliable:-** Message integrity is checked using SHA-512 secure hash function.

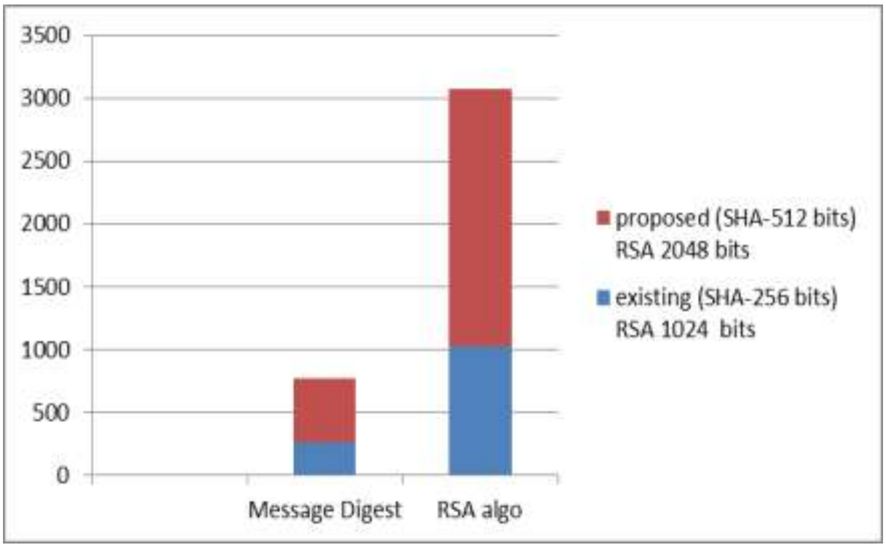


Figure 12: Cryptographic algorithm comparison

8. **Fast transfer:-** The proposed system used the concept of OOPs where information is transferred in the form of object instead of streams. All the information is encapsulated in a object which is securely send from one end to another.

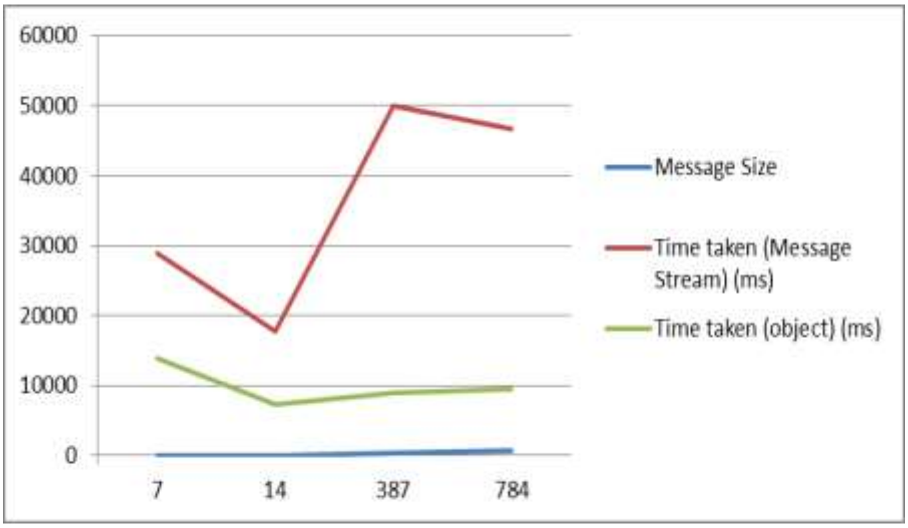


Figure 13. Time Comparison between Object and Stream for Different Messages

5. Conclusion

PKI provides strong authentication, confidentiality, privacy, integrity, non-repudiation and tamper detection via set of well-established techniques and standards. It manages private public keys in unsecure environments for electronic transactions. Certificates are required for safe online transactions to protect data from unauthorized users. The proposed application depicts the hierarchical flow of trust in the network. Currently PKI is the only technology that provides data integrity and protection to support secure ecommerce. This application is depicting the hierarchal flow of trust in the network. It is a multi-threading application where users of one domain belonging to different organization unit can develop a level of trust on each other. And proposed scheme is also secure against private key attack and public key replacement attack as sensitive information are stored in encrypted blob form.

The proposed algorithm satisfies all the properties required for security. It provides confidentiality, authentication and non-repudiation.

- Confidentiality is achieved by hybrid cryptography with object oriented programming where all the information is encapsulated within a object. No one those don't have knowledge about the structure of class will be able to see the information.
- Authentication is guaranteed by having the signature after identity verification by a trusted authority.
- Users can't deny for the signature as time stamp, digital signature and their identity prove their authenticity.

The hierarchical trust model designed by us is secured than the existing one. We have created digital certificate for 2048 bits and use the hybrid approach for generated the digital signature. As use of encryption and then storage of java objects in BLOB enables confidentiality and message integrity. Hence, the system will enable secure communication and provide proper authentication.

References

[1] M. Nandhini, "An Implementation of Public Key Infrastructure Using Wireless Communication Networks." International Journal of Grid and Distributed Computing, vol. 8, no. 3, **(2015)**, pp. 35-42.

[2] A. Jøsang, "PKI trust models." *Theory and Practice of Cryptography Solutions for Secure Information Systems*, **(2013)**, pp. 279-301

[3] H. Kharche and D. Singh Chouhan, "Building trust in cloud using public key infrastructure", International Journal of Advanced Computer Science and Applications, vol. 3, no. 3, **(2012)**.

[4] S.F. Al-Janabi and A. K. Obaid, "Development of Certificate Authority services for web applications", Future Communication Networks (ICFCN), 2012 International Conference on IEEE, **(2012)**.

[5] N. Vatra, "Public Key Infrastructure for public administration in Romania", Communications (COMM), 8th International Conference on IEEE, **(2010)**.

[6] H. Liping and S. Lei, "Research on trust model of pki."*Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference*, vol. 1, **(2011)**.

[7] H-C. Kim, *et al.*, "A design of one-time password mechanism using public key infrastructure", Networked Computing and Advanced Information Management, NCM'08, Fourth International Conference, vol. 1, **(2008)**.

[8] S. R. Subramanya and K. Y. Byung, "Digital rights management", *IEEE Potentials*, vol.25, no. 2, **(2006)**, pp. 31-34.

[9] A. Jancic and M. J. Warren, "PKI-Advantages and Obstacles", *AISM*, **(2004)**.

[10] S. Ten Have, E. Marcel Vander and W. Ten Have, "Key management models", Financial Times Prentice Hall, **(2003)**.

[11] S. Choudhury, K. Bhatnagar, and W. Haque, "Public Key Infrastructure Implementation and Design", M&T Books, **(2002)**.

[12] J. Weise, "Public key infrastructure overview", Sun BluePrintsOnLine, **(2001)**.

[13] R. Hunt, "PKI and digital certification infrastructure", Networks, Proceedings.Ninth IEEE International Conference, **(2001)**.

[14] J. Linn, "Trust models and management in public-key infrastructures", RSA Laboratories, vol.20, **(2000)**.

[15] C. Gaya, "Public Key Infrastructure-A Brief Overview", SANS Institute, **(2000)**.

[16] R. Perlman, "An overview of PKI trust models." Network, IEEE, vol. 13, no. 6, **(1999)**, pp. 38-43.
[17] https://www.giac.org/paper/gsec/286/public-key-infrastructure-overview/100867

Authors

Sarvesh Tanwar, she is Research Scholar, Department of Computer Science & Engineering in College of Engineering & Technology (CET), Mody University of Science &Technology, Lakshmangarh (Rajasthan), India. She received her M.Tech Degree from MMU, Mullana with Distinction and doing Ph.D from MUST, Lakshmangarh. Her research areas are Cryptography and Ad hoc networks. She has 11 years of teaching experience.

Anil Kumar, he is Professor and Head, Department of Computer Science & Engineering in College of Engineering & Technology (CET), Mody University of Science &Technology, Lakshmangarh (Rajasthan), India. His research areas are Network Security, Mobile Computing, and Big Data. He has published 200 research papers in National/International Journals/Conferences. He has guided 7 PhD students and 8 are undergoing. He is Sr. Member of IEEE , ACM and CSI. He has around 17 years of teaching and 5 years of industry experience.