A Novel Framework for Efficient Multiple Signature on Certificate with Database Security

Sarvesh Tanwar
Amity University
Sumit Badotra (summi.badotra@gmail.com)
Lovely Professional University https://orcid.org/0000-0003-1950-5386
Ajay Rana
Amity University

Research Article

Keywords: Public Key Infrastructure, Certification Authority, Man in the Middle Attack (MITM), Digital Signature, Multiple Signature, Timestamp, Aadhar Number.

Posted Date: December 16th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-794772/v1

Abstract

PKI gives undeniable degree of safety by transferring the key pair framework among the clients. By constructing, a PKI we combine digital identities with the digital signatures, which give an end-to-end trust model. Basically, PKI is an attempt, which can simulate the real-world human analyzation of identity and reliability in a computerized fashion. In any case, the existing applications are centered on a tight trust model which makes them inadequate as an overall device for trust examination. After years of research, development and deployment, PKI still facing strong technical and organizational challenges such as attacks against Certificate Authorities (CA). CAs are the primitive component of PKIs which plays powerful role in the PKI model. CA must be diligent, creditable and legitimate. In any case, a technocrat who picks up control on a CA can use CA's certificate to issue bogus certificate and impersonate any site, such as - DigiNotar, GobalSign, Comodo and DigiCert Malaysia. In this paper we proposed an approach to reduce the damage of compromised CA/CA's key by imposing Multiple Signatures (MS) after verifying/authenticating user's information. One single compromised CA is not able to issue a certificate to any domain as multiple signatures are required. Private key and other perceptive information are stored in the form of object/blob. Without knowing the structure of class no one can access the object and object output stream. Proposed MS achieve better performance over existing MS schemes and control fraudulent certificate issuance with more database security. The proposed scheme also avoids MITM attack against CA who is issuing certificate to whom which is using the following parameters such as identity of Sender, Receiver, Timestamp and Aadhar number.

1 Introduction

The fast development in e-administration applications raises a significance on the requirement for security and authenticity of the application. Therefore, various evolving technologies are trying to improve and enhance security necessities. One of the major setback that we face today in the transactions made in e-governance is replacing manual signature with an e-signature. Digital Signature can be introduced as such an electronic confirmation and there are numerous advancements, which can help us, accomplish it. Now days we are provided with wide-ranging technologies, that can help us in safeguarding the electronic set-up of any association. Today organizations require certificate-based security, which is considered as high-level security provided by Public Key Infrastructure (PKI).

It is a reliable technology, which is developing security means in various applications. If we want to secure the data or categorize the clients, then PKI is the utmost practical one. Thus, the idea of PKI incorporates public key cryptography, certificates and Certification Authority into network security architecture [3–4]. Certificates are playing an important role to map between the user identities and the public key [1]. It is used for secure data and proper authentication from users and computers both within and outside the organization [2]. A certificate includes identity (Common Name), associated public key, valid from, valid to date and unique identifier of signature authority [47]. It provides assurance of secure exchange of sensitive information over unsecure channels [5]. It enables its clients to maintain a level of trust by providing security services [6–7][48]. Public keys are accessible in the public key directories.

Before using one’s public key, must ensure its legitimacy by checking Certificate Revocation List (CRL) [8]. Status of certificate can be checked online using Online Certificate Status Protocol (OCSP). Management of trust is the more difficult issue when using such certificates. Public Key Infrastructure (PKI) is a structure for managing diverse public keys and certificates which are liable for providing, transferring and retracting of the Public Key Certificates (PKC) over a timid network, on the Internet. PKI empowers the clients of the networks to interchange the data with the help of the public and private key pairs which are acquired and shared via a trusted authority [2]. During an e-business and e-commerce transactions, coarse and conviction security principles are required to communicate the data safely over the internet. PKI is a security infrastructure which provides the necessary security principles in ventures. One of the key objectives of PKI is verification and validation of each contributor with the help of digital certificates [49]. The Certificate Policy (CP) provides a set of guidelines indicating the relevance of a certificate to a specific class with some prevalent safety requirements [3]. PKI enactments vary from region to region and from country to country. The consequences of distinct acts raise a number of key issues, as well as how to build confidence domains for distinct nations and areas and how to coordinate the distinct CAs to integrate them as if they were a single, consistent system [2] [45–46]. When distinct users from distinct domains want to interact, interoperability between PKIs is an extensive issue to be considered. Interoperability between PKI provides secure interconnection and cooperation between distinct PKI structures. PKI incapability is consigned over the cross-certification service, which can also be characterized as the way to authorized groups of trust among distinct CAs[4].

The main phase in developing PKI structure was to give the domains of confidence and define their boundaries. Each PKC entity has a non-disclosed government key and a personal key.

Symmetrical cryptography makes it difficult to carry the keys while strengthening confidentiality. With public key cryptography, confidentiality is not needed, but an assurance to safeguard against active assaults.

Security Issues: The following are the some of the security issues in terms of security: -

a) PKI deployment and technical challenges

PKI is responsible for technical aspects to support public key management for all organization. It duties cover the public/private keys generation and delivery to the users as well as publication, validation and revocation of public keys. After years of research, development and deployment, PKI still face strong technical and organizational challenges [1][9] such as attacks against PKI.

CAs are the critical component of PKIs which plays powerful role in the PKI model. CA must be truthful, honest and legitimate [10]. In any case, a hacker who picks up control on a CA can use CA's certificate to issue fake certificate and impersonate any site, such as - DigiNotar, GobalSign, Comodo and DigiCert Malaysia.

- Trust on CA [9]

- Certificate Revocation

These key challenges make use of PKI uncomfortable from an operational perspective.

b) Managing the Private keys

Managing security keys on smart devices is one of the key security concerns because the keys are required to encrypt the computer on the smart grid [11–12]. However, system devices often have limited storage capacity, power consumption, and bandwidth, and require an efficient and flexible key management scheme. Disclosure of the private key destroys the entire PKI security system [12]. Achieving simultaneous access to private keys and ease of use is one of the most important issues with PKI systems.

c) Attempts to obtain fraudulent certificates are on the rise

An attacker can disclose the key of the CA and execute an MITM attack to issue false certificates on behalf of the CA [13][38–40]. On August 29, 2011, someone attempted a MIMT SSL attack connecting Google clients and Google services [14–15]. Using an invalid SSL certificate issued by Diginotor, an attacker can revoke the certificate and cannot publish to Google [16]. Google Chrome clients are already protected against attacks from Chrome to recognize forged certificates. On September 3, 2011, the Dutch certification authority, the Diginotor framework, issued more than 531 fake SSL certificates after the security was broken. Many of these IP addresses are located in Iran, thus creating problems for partnerships between Como hackers and the Iranian government. The Iranian government effectively filters the Internet for dissident governments [17]. On February 20, 2015, Lenovo computers contained MITM adware that would drop HTTP connections. Encrypted web sessions on Lenovo computers were captured by pre-installed adware vulnerable to HTTP attacks [18]. This adware is called “Superfish”. It provides a self-signed HTTPS root certificate that allows clients to intercept encrypted traffic when visiting HTTP websites. “Superfish” manages Lenovo websites and issues certificates of fraud [18–20].

The two Taiwanese software companies, Realtek Semiconductor Systems and JMicron Technology Corp, use real software signing certificates to create digital signatures for Stuxnet malware and allow attackers to install malware on computer systems in Iran's nuclear production facility. Allow [20].

e) Attacks on Certificate Authorities

The CAs are sole point of inadequacy in PKI design. An attacker can attack the CA and defeat the entire PKI [39]. CA has the following attacks:

- Man-In-The-Middle attack

- False Certificates

- Phishing attacks

f) Public Key Replacement attack

The public key associated with the certificate is published in the public key directory and can be easily modified to replace the public key of another entity [21–22]. A mechanism to confirm deletion of a public key is needed to verify that the public key actually belongs to a particular user [23].

The signature scheme is also vulnerable to public key rotation attacks [22]. An attacker who overrides the verifier's disclosure here could create a legitimate mark on the signer's message without knowing the signer's personal information.

There are now more than 600 CAs in more than 50 countries [24]. The CA is a single point of failure [14] and rogue certificates can be issued with malicious intent, affecting the entire PKI [TOR, 08]. By revealing the CA's key, an attacker can issue fake certificates on behalf of the CA and perform various types of attacks such as MITM, Fishing, Heartbleed, SSL and TLS attacks [15][40].

For this reason, many researchers have proposed different approaches to certificate validation using Certificate Transparency (CT), public key pinning, notarization [10], or audit logs. The Google CT project requires the certificate holder to register in the audit log before using the CRT certificate. This is an open public structure that allows anyone to access the underlying segments that promote CT [27]. This is adequate for small organizations, but not to large organizations. Convergence [28] allows users to design a set of dynamic endorsers [29] that use network views for transmission authentication [31]. The Mozilla Web browser uses certificate transparency and public keying to advertise trusted certificates to clients [28] [32]. Linked data and pinned certificates are stored locally using a web browser [33] or stored on a remote web server [32]. Mozilla add-ons like CertPatrol are designed to freeze certificates and retrieve data from pinned certificates. Pinned certificates are trusted because they are verified by a notary [10]. Implementing this approach at scale is difficult for many organizations.

Negi [34] proposed a digital signature algorithm by separating the product of two discrete and large prime log problems. One limitation of this program is that it does not allow storing of digitally signed certificates. Wang, Bai and Hu, 2015 suggest several signature approaches to make the certification process go as long as one of the certificate authorities is not compromised. A Paper-Based Approach to Electronic Governance Electronic Signatures in India [24] Electronic signatures are stored in HSMs and smart cards can be lost.

The rest of the article is organized as follows: This section provides a useful overview for literature in the field of PCI. The solution to handle CA's MITM attacks implements multiple signatures [10] using the aadhar number [24] provided by UIDAI to authenticate the online entity prior to authentication, 3.I will explain in the part. The CA / Sub CA certificate generation algorithm is described in Sect. 3.2, and the respective certificates are reviewed and a security analysis is performed in Sect. 3.2.1 and 3.2.2. Performance analysis is performed in Sect. 4.

2. Literature Review

A PKI framework works by having a CA for issuing public key certificates which are fundamental well spring of trust in the exchange. It offers validation by means of computerized authentications, which are signed by CA. In this manner idea of PKI coordinates computerized digital certificates; Public Key Cryptography (PKC) and CA into organize security design [4][6]. It guarantees a protected technique for exchanging sensitive information over unsecured channels [26]. It is an innovation which empowers its customers to keep up a level of trust by giving security features like authentication, confidentiality, integrity and non-repudiation. PKI accommodates an advanced authentication that can distinguish an individual or an association and directory services that can store and repudiate the certificates according to the needs of the user. A PKI is likewise called a chain of trust.

Rolf [14] explained attacks against CAs. According to him countermeasures are needed in authentication and certification revocation. According to him a PKI is not only prerequisite for the PKC but also represents a security related Achilles's heel. He also analyzed log files of OCSP servers of compromised CAs. He also discussed revocation issues, authorization issues and certificate legitimating. He explained black list approach and white list approach and told white list is more appropriate to address certificate's legitimacy.

Janabi et. al. [44] said that the most imperative security services are integrity, confidentiality, authentication, and non-repudiation. The security services of framework must be defined while outlining a communication framework. A PKI is an innovation that meets these security administrations with its strategies and principles. A PKI framework works by having a CA for issuing public-key certificates. The point of their work is to outline and actualize a CA framework that can make and relegate public key certificates. Thus, the framework empowers secure communication and legitimate verification. Other than the essential security prerequisites, the created framework uses an approach that can facilitate in the revocation of the certificates. The design and implementation of their proposed system have been achieved using PHP and HTML programming languages besides Apache web server and MySQL database server.

Wang et. al. [10] recommended MS on a certificate. A compromised CA can prevent the entire PKI framework and issue fake certificates to arbitrary domains without the consent of the domain owner. Fake certificates can be used in MITM attacks. They recommended a multi-signature method for single server certificates.

Jain et. al. [24] portrayed DS structure to recognize signing of transactions for e-Govt. and non e-Govt. applications. They discussed the cloud-based DS initiatives they have taken to improve security during e-government exchanges in India. After confirming the incoming request, aadhar authentication data is sent to the e-Sign KYC service. It is always returned to the digital signature authority if valid or unsuccessful. If the e-KYC is successful, the e-Sign provider generates a key pair and a CSR, and generates a DS using a 160-bit message.

Nia et. al. [43] looked at different sorts of DS schemes in light of efficiency, security level and complexity. They explained different type of DS schemes and procedures such as batch scheme; forward secure scheme, blind scheme and proxy scheme.

H:{0,1}^*→G1

The proposed scheme satisfies all the properties required for an authentication encryption for example – privacy, validation, forward secrecy and non-denial. But this scheme experiences the key escrow and non-revocation issues.

Malone [42] utilized the idea of a cryptographic system based on identity and signature encryption. He designed a cryptographic identity scheme based on a bilinear connection. He also presented cryptographic evidence in curated cryptographic attacks to secure cryptographic systems based on oracle-type random recognition.

Zheng [22] proposed a signcryption method based on the Discrete Log Problem (DLP) in which the sender generates a symmetric (shared) key using the public key of the receiver. The scheme did not take into account the possibility of public verification, secret transmission, and authenticity of encrypted messages.

On the basis of detailed literature review on PKI; Table 1 shows the summary of literature review.

Table 1
Summary of Literature Review

Author	Year	Contribution
Albogami et. al. [55]	2021	Described how blockchain based implemented PKI is secure over traditional PKI. Analysis were done on the basis of verification, protection, reliability and execution.
Chen et. al. [51]	2021	Discussed about the keyword guess security attack in searchable encryption. Searchable engine allows the user to find their query in the encrypted information that is saved on unreliable system with guaranteed data privacy.
Samuel Lindeman et. al. [52]	2020	Designed a lightweight automate certificate enrolment protocol for IoT devices that have high constraint.
Qin et. al. [56]	2020	Presented protected upgraded design of PKI named Cecoin which distributive blockchain-based without a TTP. The scheme processed the assurance of consistency to protect from bogus certificates. Besides, it provides practical services of multi-certificates and integrity with enticing adaptability.
Chu et. al. [19]	2020	At present, generally identity verification is based on public key cryptography. Authors proposed blockchain based distributed PKI to resolved issues of rationalize PKI with the management of public key [19].
Savio Sciancalepore et al. [50]	2019	Proposed a distributed CL key agreement protocol for integration in Zigbee 3.0 and IoT devices that are highly constraint.
Singla A et. al. [38]	2018	Deployed PKI for shielding IoT devices. Authors also analylized performance, efficacy, security and scalability of the proposed approach.
Yakubov et. al. [25]	2018	Designed blockchain based PKI setup scheme to lessen certificate revocation issues and single point of breakdown due to CA misbehaviour.
Lozupone [79]	2018	Described PKI as a bunch of equipment, programming, policies and specialists to deal with a domain utilizing PKI. Authors presented the concise depiction of public and private encryption frameworks with investigation, correlations of deviated and symmetric encryption plans and benefits.
Al-Bassam et. al. [37]	2017	Demonstrated decentralized PKI framework that exploit blockchain transparency for web of trust.
Yu et. al. [30]	2017	Analysis and assessment of public key certificates for designing secure certificate over Internet.
Berkowsky et. al. [53]	2017	Explained that PKI is the cornerstone technology which enables safe exchange of information via the internet.
Hafizul Islam, S. K., et. al. [57]	2017	CL DMS without bilinear paring.

Author	Year	Contribution
Kar Jayaprakash et. al. [58]	2017	Presented formal proofs of security for signcryption which were secure against side channel, chosen message, chosen cipher and fault tolrent attack.
Wu et. al. [59]	2016	Presented revocable ID based model for PKC.
Pang, Liaojun, et. al. [60]	2016	Proposed completely anonymity based multi- receiver scheme.
Yang et. al. [78]	2015	Addressed challenges in VSN such as information discrimination, resource aware information and algorithm to compute indirect trust.
Martínez et. al. [62]	2015	Implemented multisingaure scheme using JAVA based on the DLP.
Wang et. al [10]	2015	Proposed multiple signature approach on a server's certificate as the probability of breaking multiple CAs in a short period of time is reduced significantly.
Jain et. al. [24]	2015	Described DS framework to realize digital signing of transactions for e-Govt. and non e-Govt. applications.
Nandhini M [62]	2015	Proposed a solution for the DoS attacks in PKI for different application. They simulate the proposed mechanism on NS2 simulator. The solution has only analytical proof without empirical study
Dongoh Park [63]	2015	Describes various applications of PKI such as email, payment security digital document security, and server identification.
Albarqi, Aysha, et. al. [64]	2015	Explained the demand for securing communications is increasing dramatically day by day.
Braeken et. al. [65]	2015	Extend pairing–free signcryption with multiple users.
Swapna et. al. [66]	2014	Proposed ID based MSS which provides confidentiality, unforgeability and public verification
Ray et. al. [67]	2014	Proposed MCS that should be installed at each hospital for securing handling of PHI.
Szalachowski et. al. [68]	2014	To address inefficiencies in certificate transparency, they designed and proposed PoliCert, a log based proposal that define Subject Certificate Policies(SCPs) to specify parameters such as trusted CAs, update certificates, error handling in certificate and loss of private key.
Rolf [14]	2014	Explained attacks against CAs.
Yang et. al. [21]	2014	Presented Shamir and Harn's IBS schemes are not secure. Shortcomings are found in both that prompted an absence of security around the signer's secret key. They proposed an enhanced scheme to solve the problem of attacker's knowledge of the signer's secret key.
Jøsang et. al. [20]	2013	Takes a closer look at the most important and mostly used PKI trust models and related semantic issues.

Author	Year	Contribution
Hassouna et. al. [69]	2013	Examined the shortcoming of the current mobile banking schemes based on PKI and IDC and proposed a web based mobile banking scheme based on CL cryptography.
Laura [70]	2012	Represented a signcryption framework which was based on Schnorr DS algorithm
Janabi et. al. [44]	2012	Aims to design and implement a CA to create and assign public key certificates for web application.
Reddy et. al. [71]	2011	Demonstrate simple application of PKI, CA and certificate repository using openSSL.
Domiguez et. al. [72]	2011	Implementation of RSA based efficient ID based multisignature scheme.
Fagen et. al. [75]	2011	Construct a more flexible scheme which allow ID and message of arbitrary length, collision resistance hash function and used a secure one time symmetric key encryption scheme.
Zhang et. al. [73]	2010	Proposed a novel AC-PKI framework for ad hoc networks which empower public-key services with CL public keys so that the complications regarding certificate management can be avoided which are inevitable in conventional certificate-based solutions.
Durán Díaz, Raúl, et al. [74]	2010	Review multiple signatures with their pros and cons.
Sharmila deva [76]	2010	Described an ID based cryptosystem serve as an efficient alternative to PKI.
Xie et. al. [77]	2010	Presented pairing free CL signcryption scheme.

3. Proposed Solution

Certificates which are sealed by a CA tend to be modified during the certification process. After reviewing the material, look for work possibilities. In [10], the author recommends multiple signatures on certificates, but believes that it is more secure to apply Aadhar authentication to multiple signatures. It also uses 512-bit message digest instead of 160 bit compared to [24].

Today, the Aadhaar has become an important document not only to verify identity, but also to help society, organizations and governments in case of financial troubles. The proposed solution would be implemented in Java using a database connection and Jcreator in Xampp. Certificates are designed to be signed by multiple CAs only after Aadhar validation on the server to dramatically reduce the time to issue fraudulent certificates. Our approach also handles DoS attacks when validating information using time stamps. Our solution does not allow you to issue a certificate to the server if the certification authority is attacked because the database or the key requires multiple signatures and Aadhar [35–36].

3.1 Multiple Signature Generation Scenarios

After validating / authenticating user information, we proposed an approach to force multiple signatures to reduce compromised CA / CA key damage. A compromised CA cannot issue a domain certificate because it requires multiple signatures.

3.2 Algorithm for Certificate Generation for CA/ SubCA

A certificate will be sealed by numerous CAs instead of single CA [10]. Multiple signatures are imposed on a server certificate. Such as Server Certifice issigned by $SUBCA01, SUBCA12andCA00$. Certificate has numerous paths. Client verify multipath after affirmation of certificate, client can determine to do transmission or not.

Step 1: User creates user account with user ID, password and security code.

Step 2: CA/SubCA send certificate request by giving his information to RootCA/CA.

Step 3: RootCA/CA check request for certificate, then verifying all the information from UIDAI server by specifying Aadhar number. If all the information matched then processes the request otherwise ignore that.

```
If Aadhar_no Verify
Then Process
else
Ignore
```

Step 4: After certificate generation secret message is send to CA/SubCA to verify certificate is issued and received by authentic entities.

```
//Message send by CA
final byte[] cipherText1 = se.encryptData(data1,pubkey,"RSA/ECB/PKCS1 Padding");
hobj.setMsg(cipherText1);
toClient.writeObject((HandShake)hobj);
System.out.println("received message from client ");
//Message decrypt by SubCA
byte []bb = se.decryptData(hobj.getMsg(),privkey,"RSA/ECB/PKCS1Padding");
```

ByteArrayInputStream bais1 = new ByteArrayInputStream(bb);

Step 5: END

Steps for Certificate Generation for Server/Client

Step 1:

Certificate Request SubCA

Step 2:

SubCA check request for certificate, then verifying all the information from UIDAI server by specifying$A$ adhar number. If all the information matched, then processes the request otherwise ignore that.

$$IfAadhar\_noVerify$$

$$ThenProcess$$

$$else$$

$$Ignore$$

Step 3:

SubCA sign the certificate and Send certificate request of other CA’s to sign the certificate with multiple signatures.

$$//Signature1$$

$$finalbyte[]sign1 = encrypt(msgdgstbyte, privkey);$$

$$ByteArrayInputStreambar = newByteArrayInputStream(sign1);$$

$$ObjectInputStreaminpr = newObjectInputStream(bar);$$

$$Stringdsign1 = (String)inpr.readObject();$$

$$oo.setSign1(dsign1);$$

Step 4:

CA send certificate to SubCA by encrypting request with SubCA’s public key.

$$finalbyte[]certenc = encrypt(cert1, pubkey1);$$

Step 5:

Only SubCA can open that certificate, by decrypting with his private key.

$$final byte[] certenc1 = decrypt(cert1, privkey);$$

Step 6:

$$OtherCA' s sign the certificate and send back to SubCA$$

$$//Signature2$$

$$byte[] msgdgstbyte = bos.toByteArray();$$

$$final byte[] sign1 = encrypt(msgdgstbyte, caprivatekey);$$

Step 7:

SubCA send certificate to user.

This process will be repeated n times, if number of signer is n.

3.2.1 Results:

Figure 4 shows the certificate issued to RCA

Figure 6: User Request for a Certificate

As certificate received by the user/sub-ca acknowledgement will be sent to CA/RCA. For this there will be a handshaking process encrypted with shared key and a timestamp for the validity of the certificate.

.

Figure 7 : Handshaking between TS and CA

Figure 9 (b): Root CA Repository

Figures 8 and 9 shows how information is stored in database. Private Key and sensitive information are stored in blob form for the security.

3.2.2 Certificate Verification

In order to process the certificate path, verifier should verify the validity of the certificate in certificate path. For certificate validation one should check the legitimacy of the signature over certificate content. The public key of the verifier is used to check the signature. However, a certificate having a valid signature does not have sufficed to qualify that certificate is valid because there may be some extra personal privacy requirements related to trust.

Server Certifice signed by $SUBCA01$, $SUBCA12$ and $CA00R$ would have the certificate path

$$ServerCA001 SUBCA011,$$

$$ServerCA002SUBCA012,$$

$$ServerCA00R$$

Signature verification process

- On the M it then computes the unique fingerprints with the help of similar hash algorithm which is applied as in CA_i .
- Then decrypts the encrypted value of hash or signature with the support of the public key ofCA_i.
- Afterwards, Compares the values of decrypted value of signature and hash which is computed by the verifier. If those two matches, then the message will be signed by CA_i and the value of message is untampered. Figure 10 below shows signature verification process for the same.

This method takes time to verify the certificate because the CAs are organized hierarchically. If PL is the length of the path, the certificate validation can be expressed as-

$O\left(PL^i\right)$where$i$is number of CAs

The Path length complexity can be reduced by summarizing the domain CA's who are responsible to issue certificates and verifies the signatures on a certificate for a particular domain. Figure 11 shows the following that how verification is done by the multiple CAs. Client first of all verifies the server certificate. For this it will check its certificate path as well. It has three certificate paths:-

$$CA001SUBA01Server$$

$$CA002SUBCA012Server$$

$$CA00RServer$$

Client put request to$CA001$, $CA002$ and $CA00R$ to verify it by using their public keys.

3.2.3 Security Analysis

For the proposed algorithm, the main concern is the security. It meets all the predefined attributes that are required for the security, confidentiality, integrity of authentication, and non-repudiation. The proposed algorithm is the main constraint for which the identity is verified by the trusted server. In the India Government Authorities there are several initiatives with regards to the Digital India Program initiatives which enables the e-Government transactions in online mode. For securing such transactions, the Government of India has been taken various steps one of them is the e-Sign [35]. With this the Government of India has also introduced the biometric authentication for the self-identification of the person which helps in reducing the frauds in the public distribution figure helps in enabling the system of government to serve better for the nation [35–36]. This concern of security then came up creating the Unique Identification Authority of India knows as UIDAI. Here the biometrics which provides a unique identification number is called as Aadhar Card number which is uniquely generated for each and every individual separately. This number will then be linked to the all-financial institutions such as gas stations, academic institutes, financial institutes and to the PAN income tax offices.

Also, this can be checked by the passport distribution authority while issuing the passport for the verification purposes before issuing it. This is where you can use your Aadhar number and the details associated to it just as email-ids and the unique identifier which can authenticate with UIDAI using the KYC processing service. This can be useful to checking if someone has forged so that you can directly check the UIDAI server for the same.

Let there are N CAs in the system that required S signatures on a certificate. The number of compromised CAs on a short period of time can be T_c and Sn denotes number of signatures on a certificate. In a hierarchy a root CA, sub CA or user can be compromised. A certificate signed by multiple CAs with aadhar authentication reduces the damage of breached CA. The security of the proposed approach can be analyzed in the following scenarios:

- $T_c cript >$: The adversaries will not be able to issue a bogus certificate as it required S signatures on the certificate. The compromised CA cannot bring damages to the network communication.
- $T_c = Sn$: The probability of using rogue certificate for any domain at a short period of time (PT_{Tc}) can be expressed as follows:

$$PT_{Tc} = Pc^{Tc} = Pc^{Sn}$$

Where PT is the probability of compromised one CA. For example if Pc=3%=0.03 and Sn=4 then $PT_{Tc} = 0.03^4 = 0.00000081 = 0.000081\%$. Probability of four compromised CA's key is less than one compromised CA's key.

- **Database security**: Our proposed algorithm is providing more database security over [10][24]. Authors have not given any information how sensitive information is stored. Our main emphasis is on database security as well as efficient certificate issuance. Private key and other information is stored in the form of object/blob [41]. Without knowing the structure of class no one can access the object. This is the main advantage over the existing certificate issue algorithm.
- **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 [31] 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 [35]. Database of private key storage is shown in Fig. 12.

- **Strong authentication**: Here fabrication of identity is avoided. Now a day if you are taking Sim cards that is also registered with aadhar number. By specifying your aadhar number or fingerprints subscriber is able to check user's biological information. As ID's of user's are also verified by UIDAI this will also avoid MITM attack who is issuing certificate to whom.

$$M = ID_A \parallel ID_B, Info_A + N_1, Aadhar\_no$$

- **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. A public key directory is presented in Fig. 13.

4. Performance Analysis

The proposed system is more secure in term of cryptographic operation. Tables 2, 3 and 4 show the performance analysis of the proposed approach.

Phases: Existing approaches has following phases: *registration, signature generation and verification*. But in proposed approach, we also add one new phase e.g aadhar check. After registration aadhar check will be performed by the signer via KYC services that not only check the identity information but also compare biometric information. Sometime an adversary changes the identity and impersonates himself as legitimate person, which is very difficult to trace him. But in aadhar authentication as biometric information is scanned, all the information of that person will be checked on the basis of that biometric information such as finger print, iris. Security of Multiple Signatures: Multiple signatures are very strong as compared to batch and blind signature. Table 3 represents the comparison between different signatures.

Table 2
Number of Cryptographic Operations

Name of phase	Crypto operations
Registration	1-encryptions, 1-decryptions
Aadhar Check	2-encryptions, 2-decryptions and checking of digital signature
Multiple Signature	n-encryption, checking of digital signature and 1-decryption

Table 3
Comparison between Digital Signature Schemes

Bases	Batch	Blind	Multiple Signature
Security	Middle	Strong	Very Strong
Efficiency	Average	Very efficient	Very Efficient
Verification	Middle	Good	Good
Difficulty	Low	Middle	Average

Table 4
No. of Operation for Signature Generation

Name of phase	Crypto operations
Registration	1-encryptions, 1-decryptions
Digital Signature	1-hash, 1- encryption, 1 decryption
Multiple Signature (No. of signature = 2)	3- hash, 4-encryption, 4- decryption

- **Number of operations**: As multiple signatures imposed on certificate. Table 4 shows how many operations are required if number of signer is 2.
- **Cryptographic Algorithm**: Earlier signature used RSA 1024 bits with SHA 256 bits hashing algorithm. But we use all the latest version of cryptographic algorithms such as RSA 2048 bits and SHA 512 bits message digest algorithms. Figure 15 describes comparison between cryptographic algorithms.

Fast transfer

- Existing approaches send the information in streams that takes more time in transmission. But 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 an object which is securely send from one end to another. As the size of information increase it takes less time to send and receive the messages. Table 5 explains the comparison of time between previous approaches and proposed scheme.

Table 5
Comparison of Time between Previous approaches and Proposed Scheme

Message Size	Time taken in existing approaches (Message Stream)	Time taken in proposed approach (OOPs)
(bytes)	**(ms)**	**(ms)**
7	28873	13850
14	17684	7239
387	49965	8988
784	46642	9460
11000	35568	3027
32400	26255	1872

6. Conclusion And Future Scope

PKI make use of two cryptographic keys- private and public where owner of private key can only decode the confidential information that are encoded by public key of the sender. In our proposed approach all information is stored in the form of blob in the database for security reasons. User authentication system based on smart cards and tokens have their own security vulnerabilities. Smart cards are prone to cyber-attacks. Smart card reading machines are expensive and not compatible with every smart card. Tokens can be breached during the message transmit. Even security pin can also be misplaced or forgotten. Aadhar card number is a digital identification platform that provides powerful authentication technique. It is deployed in multiple domain such as in banks, Digital Locker, gas subsidy, financial organizations, passport acquisition and for filling digital application to extract the data have to link Aadhaar, owing to its high security and privacy mechanisms. The transmission between bank and UIDAI is secured with encrypted network. Along with identity verification, digital certificate is signed by numerous CA’s making it unfeasible to corrupt all the CA’s at once. Despite that Aadhaar authentication is a complicated procedure, it involves foremost cryptographic algorithms making it highly secure as compared to those that are currently available. Providing security is the top most priority so overhead can be compromised. In our proposed algorithm, CA’s identification is validated by RCA and User’s identification is validated through RA’s and CA’s from UIDAI server respectively. It consists of confidential data and biometric information which can prevent identity fabrication. Biometric information is encoded with PKI. It mitigates security concern by providing digital certificates which act as an identity proof with a limited lifetime. The proposed system also used the concept of OOPs where information is transferred in the form of object instead of streams. All the information is

encapsulated in an object which is securely send from one end to another. Certificates are designed to be signed by multiple CAs only after aadhar validation on the server to dramatically reduce the time to issue fraudulent certificates. Our approach also handles DoS and MITM attacks when validating information using time stamps. It does not allow to issue a certificate to the server if the CA is attacked because the database or the key requires multiple signatures and aadhar authentication. In future traditional PKI system can be implemented with blockchain to eliminate the single point of failure of CAs.

Declarations

Funding Information

This work is not funded by any agency

Conflict of Interest

Authors declare no conflict of interest

Availability of data material and code availability

Authors declare that the research work has no data availability material and code availability

Authors' Contributions

Dr. Sarvesh Tanwar has performed the experimentation and carried out the research. Dr. Sumit Badotra assisted Dr. Sarvesh Tanwar and helped in writing the paper. Dr. Ajay Rana formatted the paper and checked for proof reading.

References

1 Jancic, A., and Matthew J. Warren, “PKI-Advantages and Obstacles”, *AISM*, pp. 104-114, 2004.

2 Jarupunphol, Pita, and Chris Mitchell, “PKI implementation issues in B2B e-commerce,” In *EICAR Conference Best Paper Proceedings, UE Gattiker, Ed. Copenhagen, pp. 1-14,* 2003.

3 Adams, Carlisle, and Steve Lloyd, “*Understanding public-key infrastructure: concepts, standards, and deployment considerations”,* Sams Publishing, 1999.

4 Choudhury, S., K. Bhatnagar, and W. Haque, *“Public Key Infrastructure: Implementation and Design”, Hungry Minds M&T Books, New York, NY* 10022, 2002.

5 Wilson, Stephen, The importance of PKI today”, *China Communications*, pp. 15-21, 2005.

6 Vatra, Nicusor, “Public key infrastructure for public administration in Romania”, *Communications (COMM), 8th International Conference*, *IEEE*, pp. 481-484, 2010.

7 Sharick, T. M., J. P. Long, and B. J. Desind, “Development of a public key infrastructure across multiple enterprises”, *Sandia National Labs.,* Albuquerque, NM (United States), pp. 1-7, 1997.

8 Choudhury, S., K. Bhatnagar, and W. Haque, *“Public Key Infrastructure: Implementation and Design”, Hungry Minds M&T Books, New York, NY* 10022, 2002.

9 Ellison, Carl, and Bruce Schneier, “Ten risks of PKI: What you're not being told about public key infrastructure”, Computer *Security Journal,* vol.16 (1), pp, 1-7, 2000.

10 Wang, Xinli, Yan Bai, and Lihui Hu (2015). Certification with Multiple Signatures. *In Proceedings of the 4th Annual ACM Conference on Research in Information Technology*, ACM,pp. 13-18.

11 Alohali, Bashar, Kashif Kifayat, Qi Shi, and William Hurst, “A survey on cryptography key management schemes for smart grid”, *Journal of Computer Sciences and Applications,* vol. 3(3A), pp. 27-39, 2015.

12 Barker, Elaine, William Burr, Alicia Jones, Timothy Polk, Scott Rose, Miles Smid, and Quynh Dang, “Recommendation for key management part 3: Application-specific key management guidance”, *NIST special publication,* vol. 800(57), 2009.

13 Huang, Lin Shung, Alex Rice, Erling Ellingsen, and Collin Jackson, “Analyzing forged SSL certificates in the wild”, *Security and privacy (sp), 2014 IEEE symposium*, *IEEE*, pp. 83-97, 2014.

14 Oppliger, Rolf, “Certification authorities under attack: A plea for certificate legitimation”, *IEEE Internet Computing,* vol. 18(1), pp. 40-47, 2014.

15 Netcraft, Certificate authorities issue SSL certificates to fraudsters, 2015

16 EFF, “iranian-man-middle-attack-against-google”, Deeplinks/2011/08, 2011.

17 Kirk, J. “Comodo hacker claims credit for DigiNotar attack.” Computerworld , 2011.

18 Kenn White, “Lenovo computers come with pre-installed adware and MITM proxy”, Helpnetsecurity, 2015.

19 Chu, Y., Kim, J. M., Lee, Y., Shim, S., & Huh, J. (2020, January). SS-DPKI: Self-signed certificate based decentralized public key infrastructure for secure communication. In *2020 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-6). IEEE.

20 Jøsang, Audun, “PKI trust models”, *Theory and Practice of Cryptography Solutions for Secure Information Systems, Scopus*, pp. 279 -301, 2013.

21 Lu, Yang, and Jiguo Li, “Efficient certificate-based signcryption secure against public key replacement attacks and insider attacks”, The Scientific World Journal, vol. 14(1), pp 1-13, 2014.

22 Zhang, Zhenfeng, and Dengguo Feng, “Key Replacement Attack on a Certificateless Signature Scheme”, *IACR Cryptology, ePrint Archive*, pp 453-457, 2006.

23 Hu, Bessie C., Duncan S. Wong, Zhenfeng Zhang, and Xiaotie Deng, “Key replacement attack against a generic construction of certificateless signature”, *ACISP*, vol. 6, pp. 235-246, 2006.

24 Jain, Vijay, Ranjan Kumar, and Zia Saquib. “An Approach towards Digital Signatures for e-Governance in India.” *Proceedings of the 2015 2nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia*, *ACM* , pp. 82-88, 2015.

25 Yakubov, A., Shbair, W., Wallbom, A., & Sanda, D. (2018). A blockchain-based pki management framework. *In The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block*) colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain 23-27 April 2018.

26 Toorani, Mohsen, and A. Beheshti, “LPKI-a lightweight public key infrastructure for the mobile environments”, *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference*, *IEEE*, pp. 162-166, 2008.

27 https://certificate.transparency.dev/, accessed on 20th June 2021.

28 http://convergence.io/details.html, accessed on 20th June 2021

29 Yüce, Emre, and Ali Aydin Selçuk, “Server Notaries: A Complementary Approach to the Web PKI Trust Model”, *IACR Cryptology ePrint Archive* 2016, 126-139, 2016.

30 Yu, J., & Ryan, M. (2017). Evaluating web pkis. In *Software Architecture for Big Data and the Cloud* (pp. 105-126). Morgan Kaufmann.

31 Jachtoma, P., B. Sakowicz, J. Wojciechowski, and A. Napieralski, “Application For Assigning Grades To Students Using Public Key Infrastructure”, In *Mixed Design of Integrated Circuits and System, MIXDES 2006. Proceedings of the International Conference*, *IEEE*, pp. 773-778, 2006.

32 Evans C., C. Palmer and R. Sleevi, “Public key pinning extenstion for HTTP”, Internet-draft, Oct. 2014.

33 Soghoian, Christopher, and Sid Stamm, “Certified lies: Detecting and defeating government interception attacks against SSL (short paper)”, *International Conference on Financial Cryptography and Data Security*, *Springer*, Berlin, Heidelberg, pp. 250-259, 2011.

34 Negi, Arvind, et. al, “New Method for Obtaining Digital Signature Certificate using Proposed RSA Algorithm”, *International Journal of Computer Applications,* vol. 121(23), pp-24-29, 2015.

35 Anil Khachi, Version 0.5. UIDAI, Lost EID/UID Process. Available at: https://uidai.gov.in/images/mou/eiduid_process_ver5_2_27052013.pdf, 2013.

36 UIDAI. UID FAQ: Aadhaar Features, Eligibility. Available at: https://resident.uidai.net.in/faqs, 2021.

37 Al-Bassam, M. (2017, April). SCPKI: A smart contract-based PKI and identity system. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* (pp. 35-40).

38 Singla, A., & Bertino, E. (2018, October). Blockchain-based PKI solutions for IoT. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)* (pp. 9-15). IEEE.

39 https://www.teiss.co.uk/what-happens-when-a-certificate-authority-is-compromised/, accessed on 10th June 2021

40 Daiki Yamakawa, Takashi Okimoto, Songpon Teerakanok, Atsuo Inomata, Tetsutaro Uehara, "Enhancing Digital Certificate Usability in Long Lifespan IoT Devices by Utilizing Private CA", Security and Communication Networks, vol. 2021, Article ID 6610863, 14 pages, 2021. https://doi.org/10.1155/2021/6610863

41 Tanwar, Sarvesh, and Anil Kumar, “A Proposed Scheme for Remedy of Man-In-The-Middle Attack on Certificate Authority”, *International Journal of Information Security and Privacy (IJISP),* vol. 11(3), pp. 1-14, 2017.

42 Malone-Lee, John, “Identity-Based Signcryption”, *IACR Cryptology ePrint Archive* 2002-eprint.iacr.org, pp. 98-105, 2002.

43 Nia, Mehran Alidoost, Ali Sajedi, and Aryo Jamshidpey, “An introduction to digital signature schemes”, *arXiv preprint arXiv:1404.2820, pp. 1-5,* 2014.

44 Al-Janabi, Sufyan Faraj, and Amer Kais Obaid, “Development of certificate authority services for web applications”, *Future Communication Networks (ICFCN), 2012 International Conference. IEEE,* pp. 135-140, 2012.

45 Singh, Priyadarshi, et al. , “Towards a Hybrid Public Key Infrastructure (PKI): A Review”, *IACR Cryptol.* ePrint Arch, Cryptology ePrint Archive: Report 2019/784, pp. 1-19, 2019.

46 Spies, Terence, “Public Key Infrastructure”, In *Computer and Information Security Handbook*, pp. 433-451. Morgan Kaufmann, 2009.

47 Goudosis, Athanasios, and Sokratis Katsikas, “ARIBC: Online Reporting Based on Identity-Based Cryptography”, *Future Internet* 13, no. 2 (2021): 53.

48 Hassouna, Mohammed, Bazara IA Barry, and Eihab Bashier, “A New Level 3 Trust Hierarchal Certificateless Public Key Cryptography Scheme in the Random Oracle Model”, Int. J. Netw. Secur. 19.4 (2017): 551-558.

49 https://sectigo.com/resource-library/what-is-x509-certificate, Jan. 2021. Accessed on 6th July 2021

50 Tedeschi, Pietro, Savio Sciancalepore, Areej Eliyan, and Roberto Di Pietro, “LiKe: Lightweight certificateless key agreement for secure IoT communications”, *IEEE Internet of Things Journal* ,7(1), pp. 621-638, 2019.

51 Chen, Zhenwei, Axin Wu, Yifei Li, Qixuan Xing, and Shengling Geng, “Blockchain-Enabled Public Key Encryption with Multi-Keyword Search in Cloud Computing”, *Security and Communication Networks*, 2021.

52 Höglund, Joel, Samuel Lindemer, Martin Furuhed, and Shahid Raza, “PKI4IoT: Towards public key infrastructure for the Internet of Things”, *Computers & Security, vol* 89, pp. 101658, 2020.

53 Berkowsky, Jake A., and Thaier Hayajneh. "Security issues with certificate authorities." In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 449-455. IEEE, 2017.

54 Khan Salabat, Liehuang Zhu, Zijian Zhang, Mussadiq Abdul Rahim, Khalid Khan, and Meng Li, “Attack-Resilient TLS Certificate Transparency”, *IEEE Access* 8 (2020): 98958-98973.

55 Albogami, O., Alruqi, M., Almalki, K., & Aljahdali, A. (2021). Public Key Infrastructure Traditional and Modern Implementation. *International Journal of Network Security*, *23*(2), 343-350.

56 Qin, B., Huang, J., Wang, Q., Luo, X., Liang, B., & Shi, W. (2020). Cecoin: A decentralized PKI mitigating MitM attacks. *Future Generation Computer Systems*, *107*, 805-815.

57 Hafizul Islam, S. K., Mohammad Sabzinejad Farash, G. P. Biswas, Muhammad Khurram Khan, and Mohammad S. Obaidat, "A pairing-free certificateless digital multisignature scheme using elliptic curve cryptography", *International Journal of Computer Mathematics,* vol. 94 (1), pp. 39-55, 2017

58 Kar Jayaprakash and Naik KSHIRASAGAR, "Security Analysis and Implementation issues of Signcryption Scheme for Smart card", A Journal of the Academy of Business and Retail Management (ABRM), vol.1(2) , pp 24-36, 2017.

59 Wu, Tsu-Yang, Jerry Chun-Wei Lin, Chien-Ming Chen, Yuh-Min Tseng, Jaroslav Frnda, Lukas Sevcik, and Miroslav Voznak, "A brief review of revocable ID-based public key cryptosystem", *Perspectives in Science*, vol. 7, pp. 81-86, 2016

60 Pang, Liaojun, Xuxia Yan, Huiyang Zhao, Yufei Hu, and Huixian Li, "A novel multi-receiver signcryption scheme with complete anonymity", *PloS one,* vol. 11(11), pp 1-18, 2016.

61 Martínez, V. Gayoso, L. Hernández Encinas, A. Martín Muñoz, and MA Álvarez Mariño, "A Java Implementation of a Multisignature Scheme", *Proceedings of the International Conference on Security and Management (SAM)*, pp. 333-339, 2015.

62 Nandhini, M, "An Implementation of Public Key Infrastructure Using Wireless Communication Networks", *International Journal of Grid and Distributed Computing,* vol. 8(3), pp. 35-42, 2015.

63 Park, Dongoh, "Social Life of PKI: Sociotechnical Development of Korean Public-Key Infrastructure" ,*IEEE Annals of the History of Computing* , vol. 37(2), pp. 59-71, 2015.

64 Albarqi, Aysha, et al, "Public Key Infrastructure: A Survey." *Journal of Information Security", vol.* 6.(1), pp. 31-37, 2014.

65 Braeken, An, and Pawani Porambage, "Efficient generalized signcryption based on ECC", *International Journal on Cryptography and Information Security, vol.* 5(2), pp. 1-13, 2015.

66 Swapna, G., and P. Vasudeva Reddy, "Efficient Identity Based Multi-Signcryption Scheme with Public Verifiability", *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 17(2), pp. 181-190, 2014.

67 Ray, Sangram, and G. P. Biswas, "A Certificate Authority (CA)-based cryptographic solution for HIPAA privacy/security regulations", *Journal of King Saud University-Computer and Information Sciences,* vol. 26(2), pp. 170-180, 2014.

68 Szalachowski, Pawel, Stephanos Matsumoto, and Adrian Perrig, “PoliCert: Secure and flexible TLS certificate management”, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, *ACM*, pp. 406-417, 2014.

69 Hassouna, Mohammed, Nashwa Mohamed, and Eihab Bashier, “A Secure Mobile Banking Scheme Based on Certificateless Cryptography in the Standard Security Model”, *International Journal of Computer Applications,* vol. 74(9) , pp. 1-6, 2013.

70 Savu, Laura, “Signcryption scheme based on schnorr digital signature”, *arXiv preprint arXiv:1202.1663, 2012.*

71 Reddy, M. I. S., Chetwavani, P. B. R., & Reddy, K. S. A Practical Approach for Implementation of Public Key Infrastructure for Digital Signatures, pp. 29-39, 2011

72 Domıguez Francisco Javier Buenasmananas and Encinas Luis Hernandez, “Digital identity-based multisignature scheme implementation”, INFOCOMP: The First International Conference on Advanced Communications and Computation, pp 42-45, 2011.

73 Zhang, Bo, and QiuliangXu, “An ID-based anonymous signcryption scheme for multiple receivers”, International *Journal of Advanced Science and Technology*, vol. 20, pp. 9-24, 2010.

74 Durán Díaz R et al (2010) A review of multisignatures based on RSA. *DIGITAL.CSIC*, pp. 1–7

75 Li, Fagen, Yongjian Liao, and Zhiguang Qin, “Analysis of an identity-based signcryption scheme in the standard model”, *IEICE transactions on fundamentals of electronics, communications and computer sciences,* vol. 94(1), pp. 268-269, 2011.

76 Selvi, S. Sharmila Deva, S. SreeVivek, and C. Pandu Rangan, “Identity based public verifiable signcryption scheme”, *International Conference on Provable Security*. Springer Berlin Heidelberg, pp. 244-260, 2010.

77 Xie, Wenjian, and Zhang Zhang, Certificateless Signcryption without Pairing”, *IACR Cryptology ePrint Archive* , pp. 187-204, 2010.

78 Yang, Qing, and Honggang Wang, “Towards Trustworthy Vehicular Social Networks”, *IEEE Communications Magazine*, vol. 53(8), pp. 42-47, 2015.

79 Lozupone, V. (2018). Analyze encryption and public key infrastructure (PKI). *International Journal of Information Management*, *38*(1), 42-44.

Figures

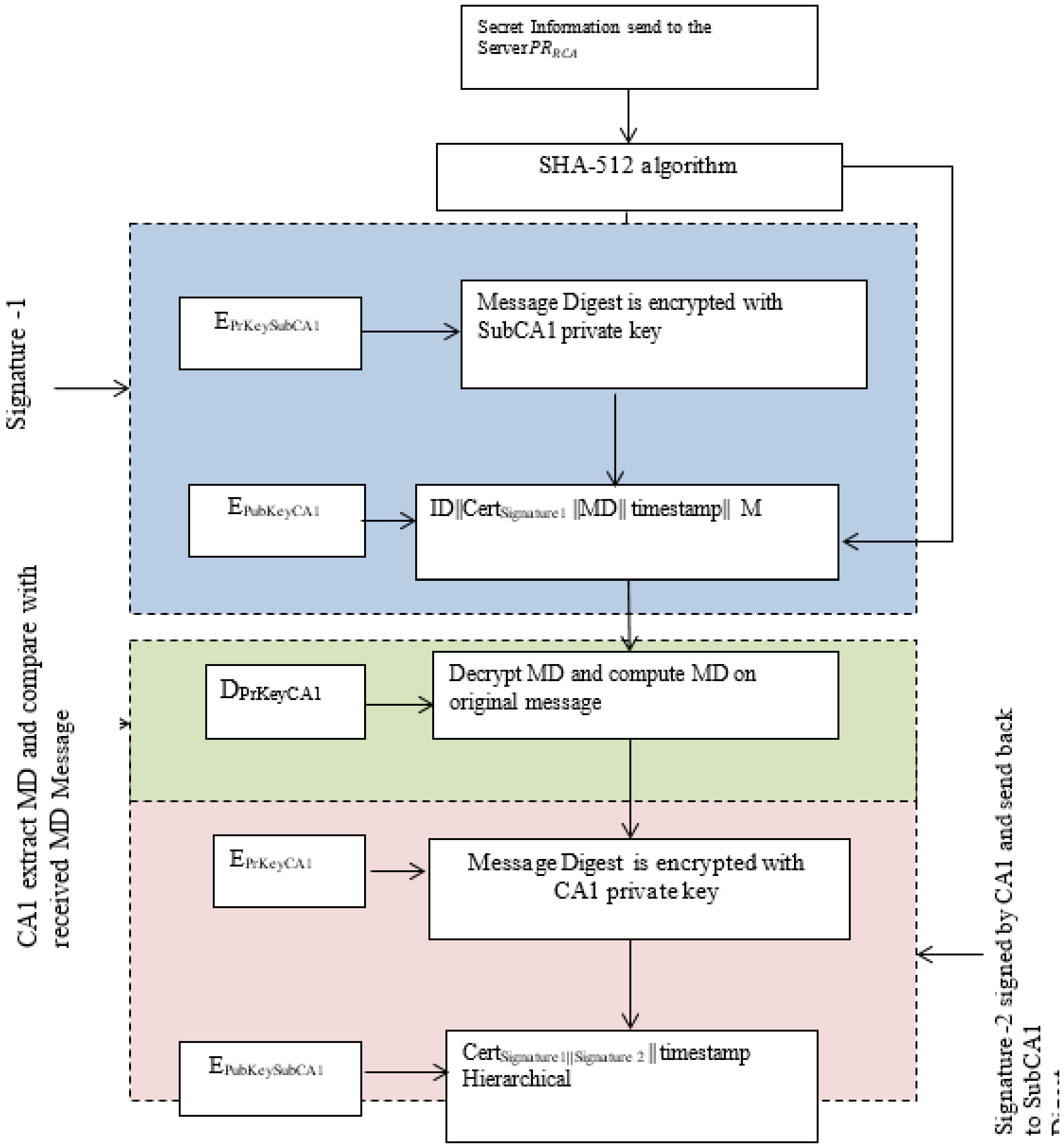

Secret Information send to the Server PRRCA
SHA-512 algorithm
Signature -1
EPrKeySubCA1
Message Digest is encrypted with SubCA1 private key
EPubKeyCA1
ID||CertSignature1 ||MD|| timestamp|| M
CA1 extract MD and compare with received MD Message
DPrKeyCA1
Decrypt MD and compute MD on original message
EPrKeyCA1
Message Digest is encrypted with CA1 private key
EPubKeySubCA1
CertSignature1||Signature 2 || timestamp Hierarchical
Signature -2 signed by CA1 and send back to SubCA1

Figure 1

Process for Multiple Signatures Generation

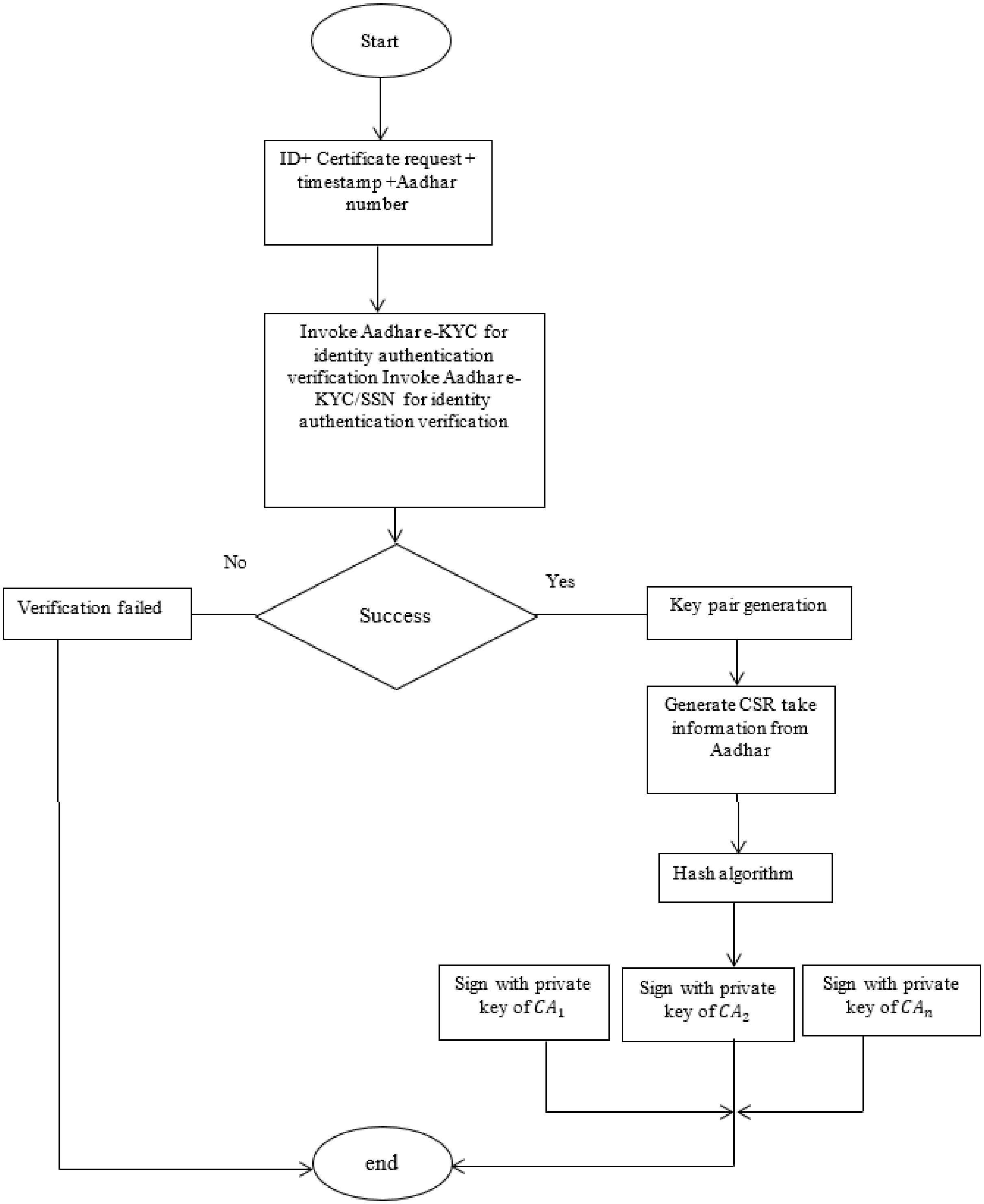

Start
ID+ Certificate request + timestamp +Aadhar number
Invoke Aadhar e-KYC for identity authentication verification Invoke Aadhar e-KYC/SSN for identity authentication verification
No
Yes
Success
Verification failed
Key pair generation
Generate CSR take information from Aadhar
Hash algorithm
Sign with private key of CA_1
Sign with private key of CA_2
Sign with private key of CA_n
end

Figure 2

Flow Chart for Multiple Signature Generation

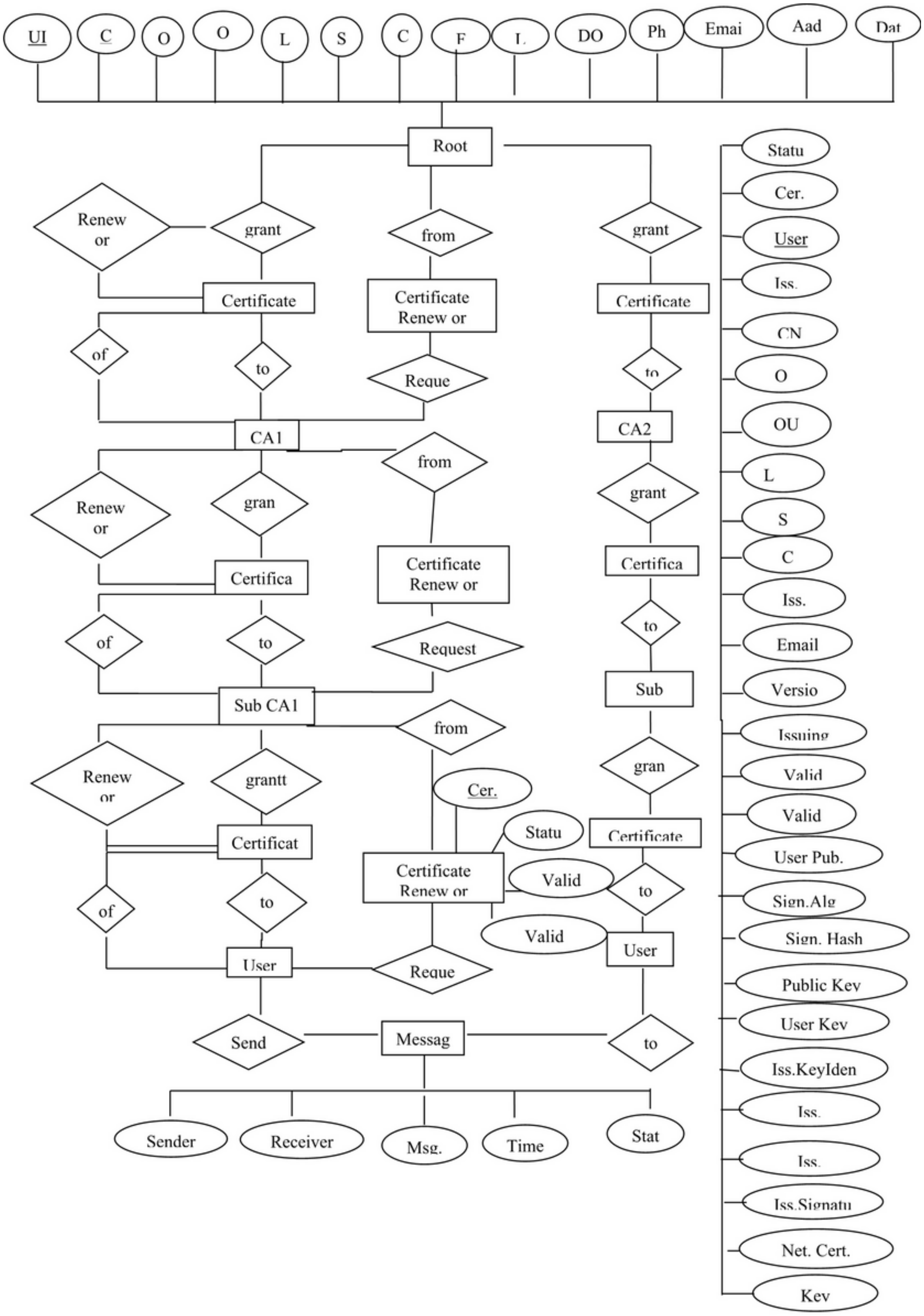
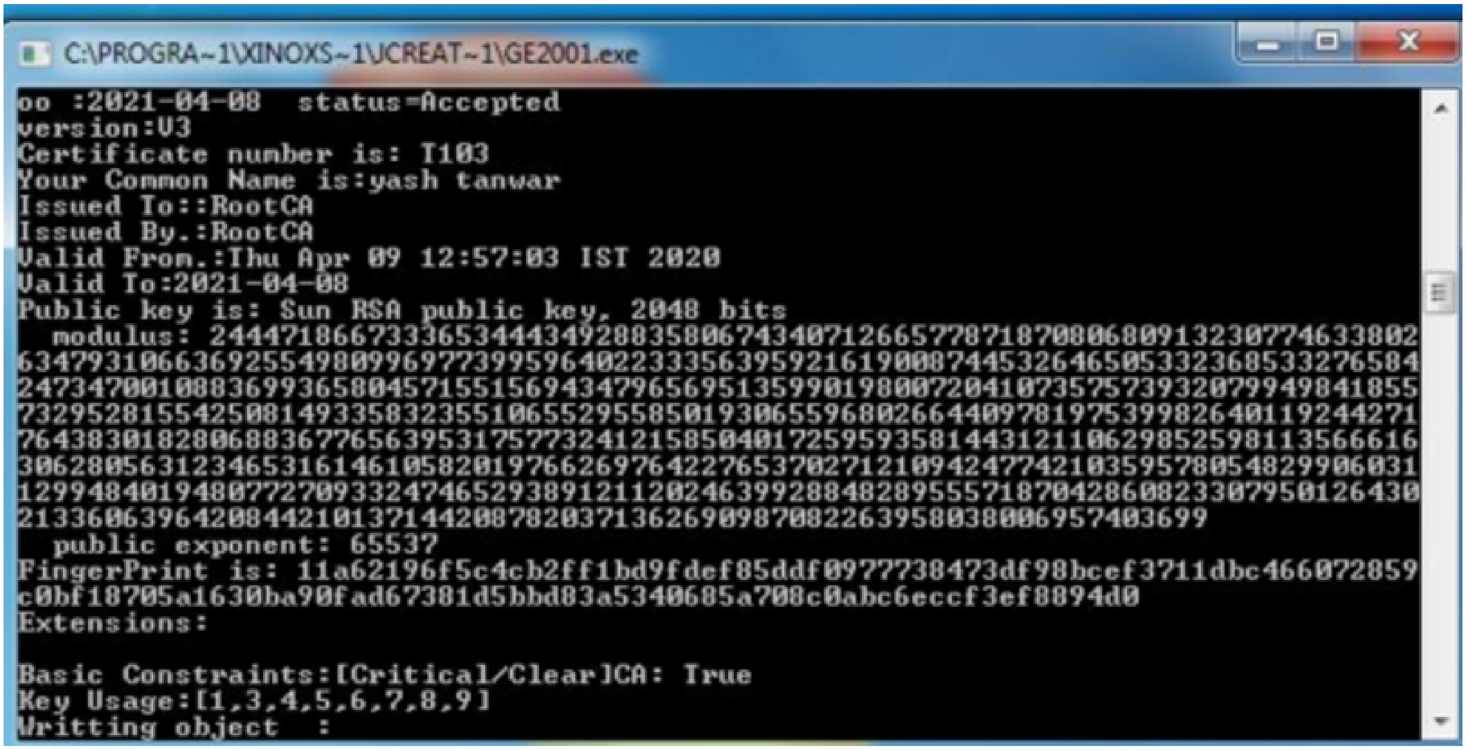


Figure 3

E-R diagram of Proposed System

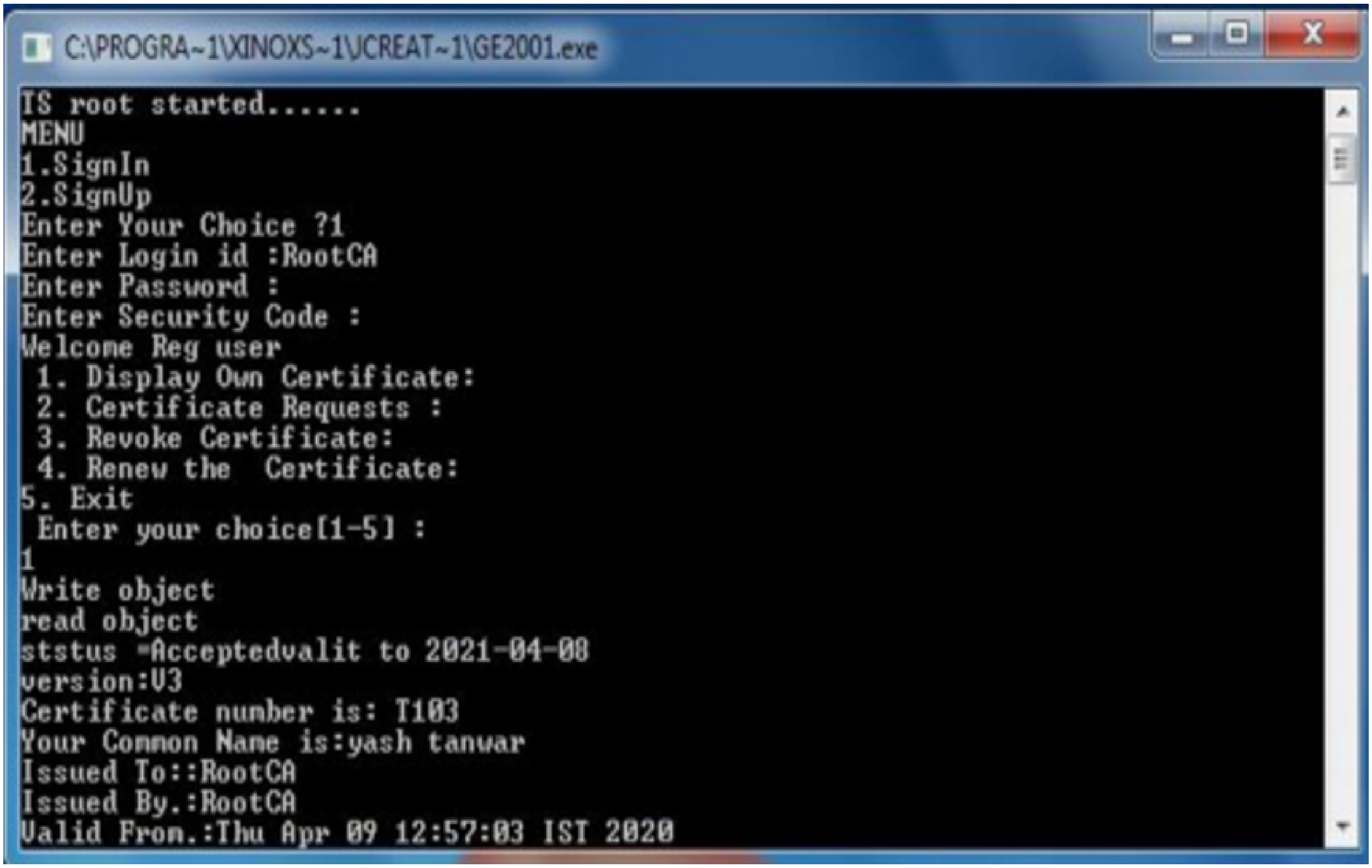


```
oo :2021-04-08  status=Accepted
version:V3
Certificate number is: T103
Your Common Name is:yash tanwar
Issued To::RootCA
Issued By.:RootCA
Valid From.:Thu Apr 09 12:57:03 IST 2020
Valid To:2021-04-08
Public key is: Sun RSA public key, 2048 bits
  modulus: 2444718667333653444434928835806743407126657787187080680913230774633802
634793106636925549809969773995964022333563959216190008744532646505332368533276584
24734700108836993658045715515694347965695135990198007204107357573932079949841855
73295281554250814933583235510655295585019306559680266440978197539982640119244271
76438301828068836776563953175773241215850401725959358144312110629852598113566616
30628056312346531614610582019766269764227653702712109424774210359578054829906031
12994840194807727093324746529389121120246399288482895557187042860823307950126430
213360639642084421013714420878203713626909870822639580380006957403699
  public exponent: 65537
FingerPrint is: 11a62196f5c4cb2ff1bd9fdef85ddf0977738473df98bcef3711dbc466072859
c0bf18705a1630ba90fad67381d5bbd83a5340685a708c0abc6eccf3ef8894d0
Extensions:

Basic Constraints:[Critical/Clear]CA: True
Key Usage:[1,3,4,5,6,7,8,9]
Writting object  :
```

Figure 4

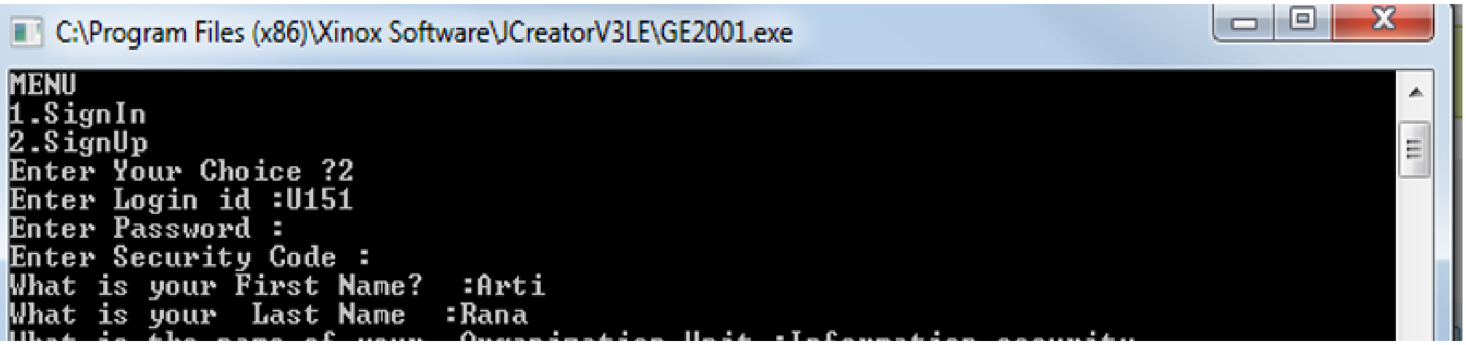
Root CA Certificate



```
IS root started......
MENU
1.SignIn
2.SignUp
Enter Your Choice ?1
Enter Login id :RootCA
Enter Password :
Enter Security Code :
Welcome Reg user
 1. Display Own Certificate:
 2. Certificate Requests :
 3. Revoke Certificate:
 4. Renew the  Certificate:
5. Exit
 Enter your choice[1-5] :
1
Write object
read object
ststus =Acceptedvalit to 2021-04-08
version:V3
Certificate number is: T103
Your Common Name is:yash tanwar
Issued To::RootCA
Issued By.:RootCA
Valid From.:Thu Apr 09 12:57:03 IST 2020
```

Figure 5

Services Provided by Root CA

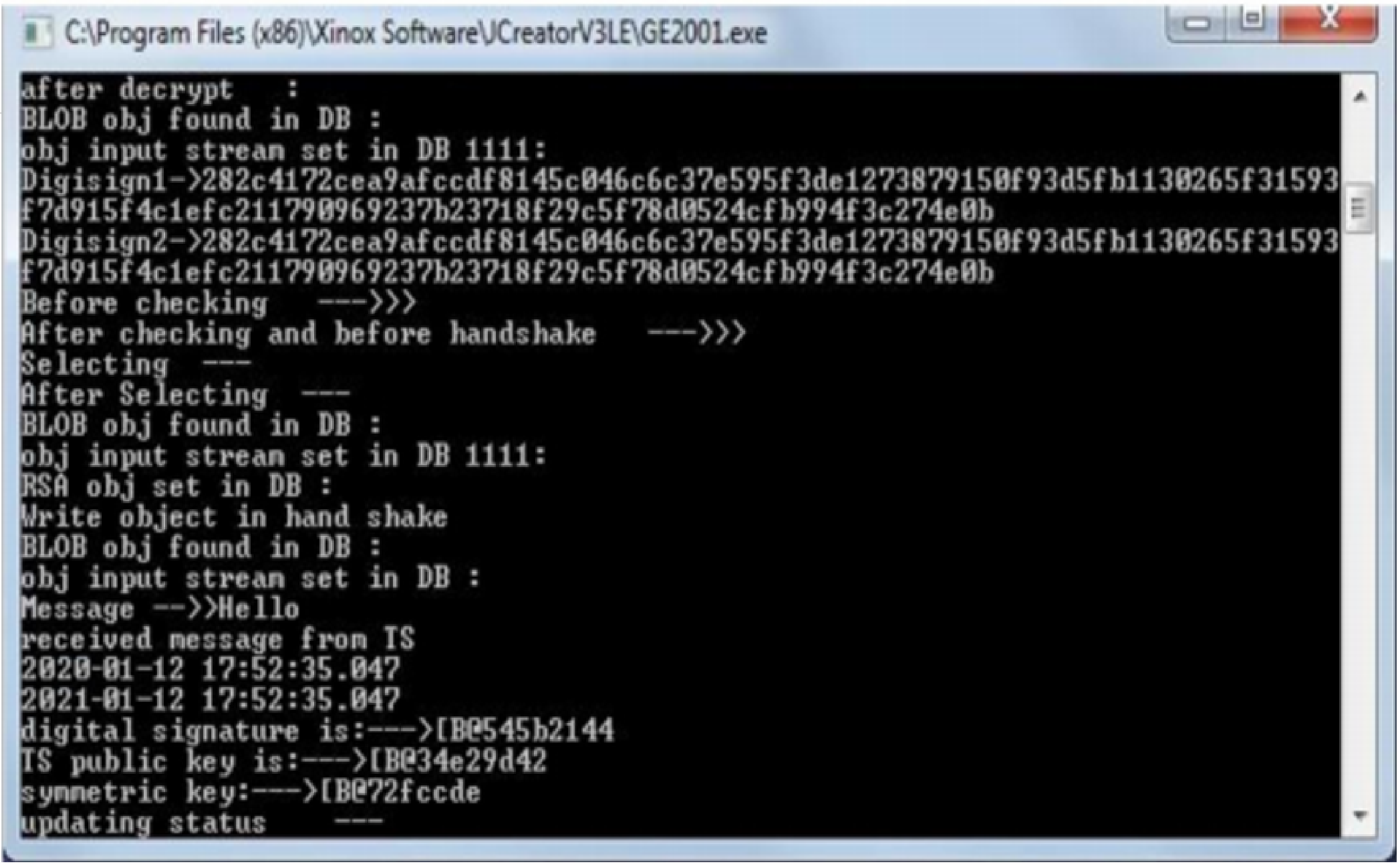


C:\Program Files (x86)\Xinox Software\JCreatorV3LE\GE2001.exe

```
MENU
1.SignIn
2.SignUp
Enter Your Choice ?2
Enter Login id :U151
Enter Password :
Enter Security Code :
What is your First Name?  :Arti
What is your  Last Name  :Rana
What is the name of your  Organization Unit :Information security
```

Figure 6

User Request for a Certificate

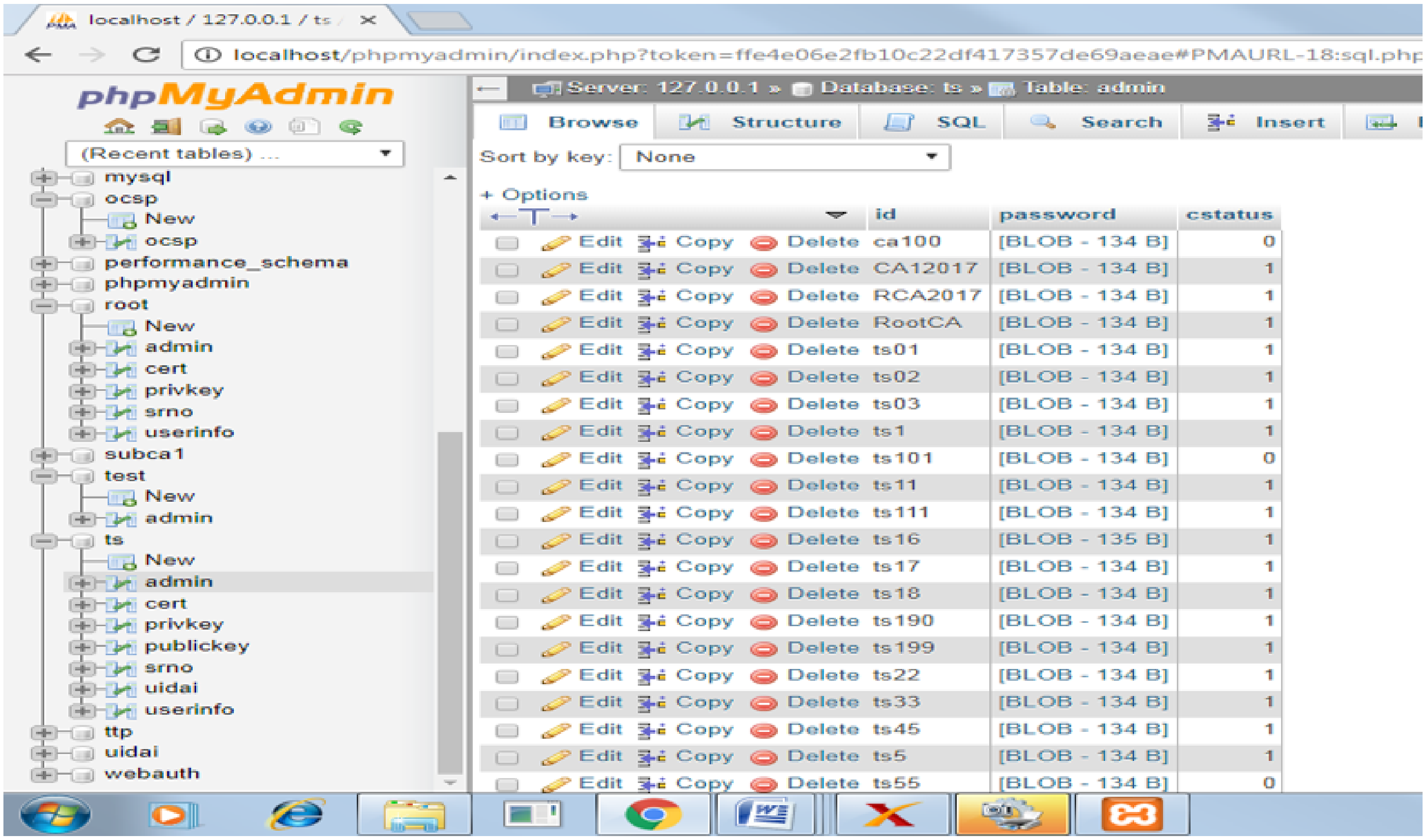


C:\Program Files (x86)\Xinox Software\JCreatorV3LE\GE2001.exe

```
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
2020-01-12 17:52:35.047
2021-01-12 17:52:35.047
digital signature is:--->[B@545b2144
TS public key is:--->[B@34e29d42
symmetric key:--->[B@72fccde
updating status    ---
```

Figure 7

Handshaking between TS and CA



localhost/phpmyadmin/index.php?token=ffe4e06e2fb10c22df417357de69aeae#PMAURL-18:sql.php

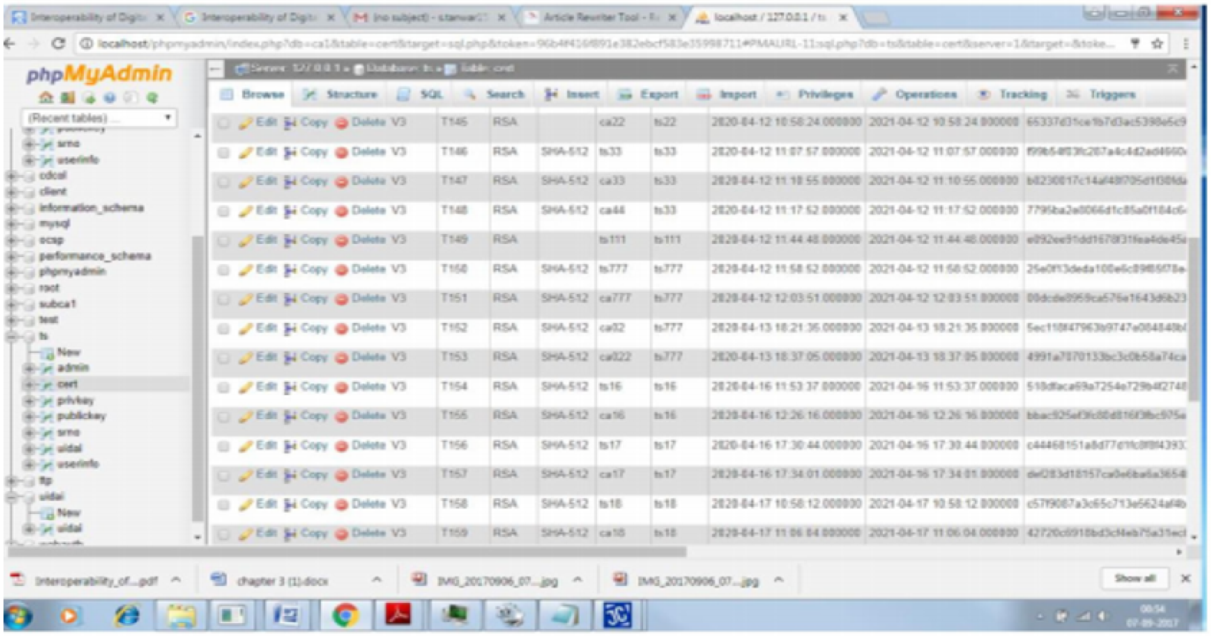
Server: 127.0.0.1 » Database: ts » Table: admin

Sort by key: None

		id	password	cstatus
□	Edit Copy Delete	ca100	[BLOB - 134 B]	0
□	Edit Copy Delete	CA12017	[BLOB - 134 B]	1
□	Edit Copy Delete	RCA2017	[BLOB - 134 B]	1
□	Edit Copy Delete	RootCA	[BLOB - 134 B]	1
□	Edit Copy Delete	ts01	[BLOB - 134 B]	1
□	Edit Copy Delete	ts02	[BLOB - 134 B]	1
□	Edit Copy Delete	ts03	[BLOB - 134 B]	1
□	Edit Copy Delete	ts1	[BLOB - 134 B]	1
□	Edit Copy Delete	ts101	[BLOB - 134 B]	0
□	Edit Copy Delete	ts11	[BLOB - 134 B]	1
□	Edit Copy Delete	ts111	[BLOB - 134 B]	1
□	Edit Copy Delete	ts16	[BLOB - 135 B]	1
□	Edit Copy Delete	ts17	[BLOB - 134 B]	1
□	Edit Copy Delete	ts18	[BLOB - 134 B]	1
□	Edit Copy Delete	ts190	[BLOB - 134 B]	1
□	Edit Copy Delete	ts199	[BLOB - 134 B]	1
□	Edit Copy Delete	ts22	[BLOB - 134 B]	1
□	Edit Copy Delete	ts33	[BLOB - 134 B]	1
□	Edit Copy Delete	ts45	[BLOB - 134 B]	1
□	Edit Copy Delete	ts5	[BLOB - 134 B]	1
□	Edit Copy Delete	ts55	[BLOB - 134 B]	0

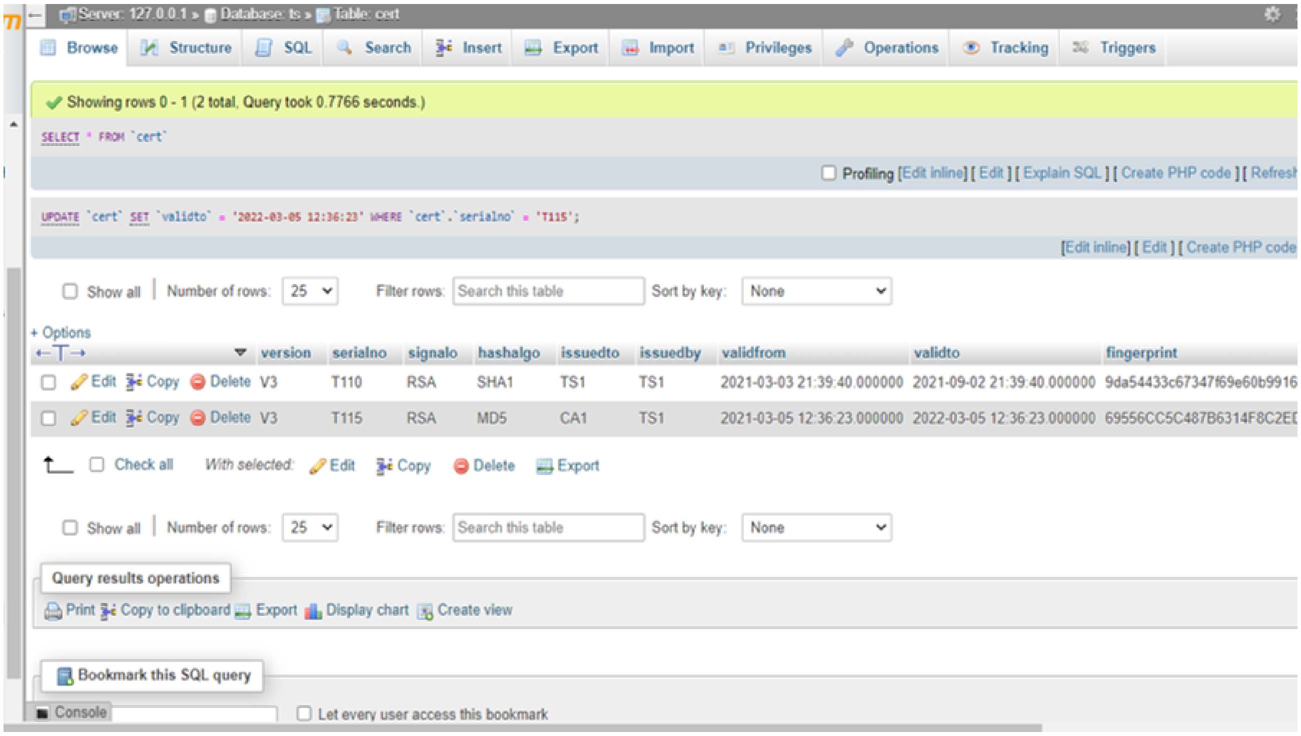
Figure 8

Database of Root CA



□ Edit Copy Delete	V3	T145	RSA		ca22	ts22	2020-04-12 10:58:24.000000	2021-04-12 10:58:24.000000	65337d31ce1b7d3ac5398e5c9
□ Edit Copy Delete	V3	T146	RSA	SHA-512	ts33	ts33	2020-04-12 11:07:57.000000	2021-04-12 11:07:57.000000	f99b54f03fc287a4c4d2ad4660
□ Edit Copy Delete	V3	T147	RSA	SHA-512	ca33	ts33	2020-04-12 11:10:55.000000	2021-04-12 11:10:55.000000	b8230017c14af48f705d1f30fda
□ Edit Copy Delete	V3	T148	RSA	SHA-512	ca44	ts33	2020-04-12 11:17:52.000000	2021-04-12 11:17:52.000000	7795ba2e0066d1c85a0f184c6
□ Edit Copy Delete	V3	T149	RSA		ts111	ts111	2020-04-12 11:44:48.000000	2021-04-12 11:44:48.000000	e092ee91dd1678f31fea4de45a
□ Edit Copy Delete	V3	T150	RSA	SHA-512	ts777	ts777	2020-04-12 11:58:52.000000	2021-04-12 11:58:52.000000	25e0f13deda100e6c89f85f78e
□ Edit Copy Delete	V3	T151	RSA	SHA-512	ca777	ts777	2020-04-12 12:03:51.000000	2021-04-12 12:03:51.000000	00dcde0959ca676e1643d6b23
□ Edit Copy Delete	V3	T152	RSA	SHA-512	ca02	ts777	2020-04-13 18:21:36.000000	2021-04-13 18:21:35.000000	5ec118f47963b9747e004840b
□ Edit Copy Delete	V3	T153	RSA	SHA-512	ca022	ts777	2020-04-13 18:37:05.000000	2021-04-13 18:37:05.000000	4991a7870133bc3c0b58a74ca
□ Edit Copy Delete	V3	T154	RSA	SHA-512	ts16	ts16	2020-04-16 11:53:37.000000	2021-04-16 11:53:37.000000	518dfaca69a7254e729b4f2748
□ Edit Copy Delete	V3	T155	RSA	SHA-512	ca16	ts16	2020-04-16 12:26:16.000000	2021-04-16 12:26:16.000000	bbac925ef3fc80d816f3fbc975a
□ Edit Copy Delete	V3	T156	RSA	SHA-512	ts17	ts17	2020-04-16 17:30:44.000000	2021-04-16 17:30:44.000000	c44468151a8d77d1fc0f0f4393
□ Edit Copy Delete	V3	T157	RSA	SHA-512	ca17	ts17	2020-04-16 17:34:01.000000	2021-04-16 17:34:01.000000	def283d18157ca0e6ba6a3654
□ Edit Copy Delete	V3	T158	RSA	SHA-512	ts18	ts18	2020-04-17 10:58:12.000000	2021-04-17 10:58:12.000000	c57f9087a3c65c713e5624af4b
□ Edit Copy Delete	V3	T159	RSA	SHA-512	ca18	ts18	2020-04-17 11:06:04.000000	2021-04-17 11:06:04.000000	42720c6918bd3cf4eb75a31ecl

a



✓ Showing rows 0 - 1 (2 total, Query took 0.7766 seconds.)

```
SELECT * FROM `cert`
```

```
UPDATE `cert` SET `validto` = '2022-03-05 12:36:23' WHERE `cert`.`serialno` = 'T115';
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	version	serialno	signalo	hashalgo	issuedto	issuedby	validfrom	validto	fingerprint
□ Edit Copy Delete	V3	T110	RSA	SHA1	TS1	TS1	2021-03-03 21:39:40.000000	2021-09-02 21:39:40.000000	9da54433c67347f69e60b9916
□ Edit Copy Delete	V3	T115	RSA	MD5	CA1	TS1	2021-03-05 12:36:23.000000	2022-03-05 12:36:23.000000	69556CC5C487B6314F8C2ED

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

b

Figure 9

(a): Root CA Repository (b): Root CA Repository

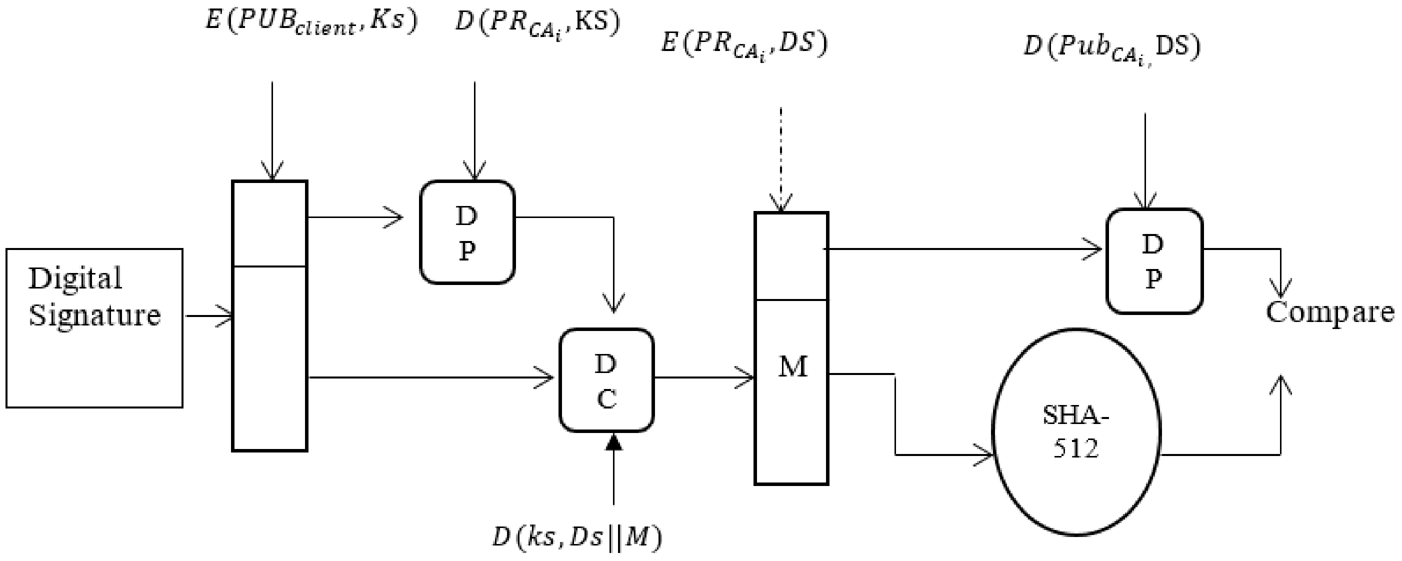


Figure 10

Digital Signature Verification

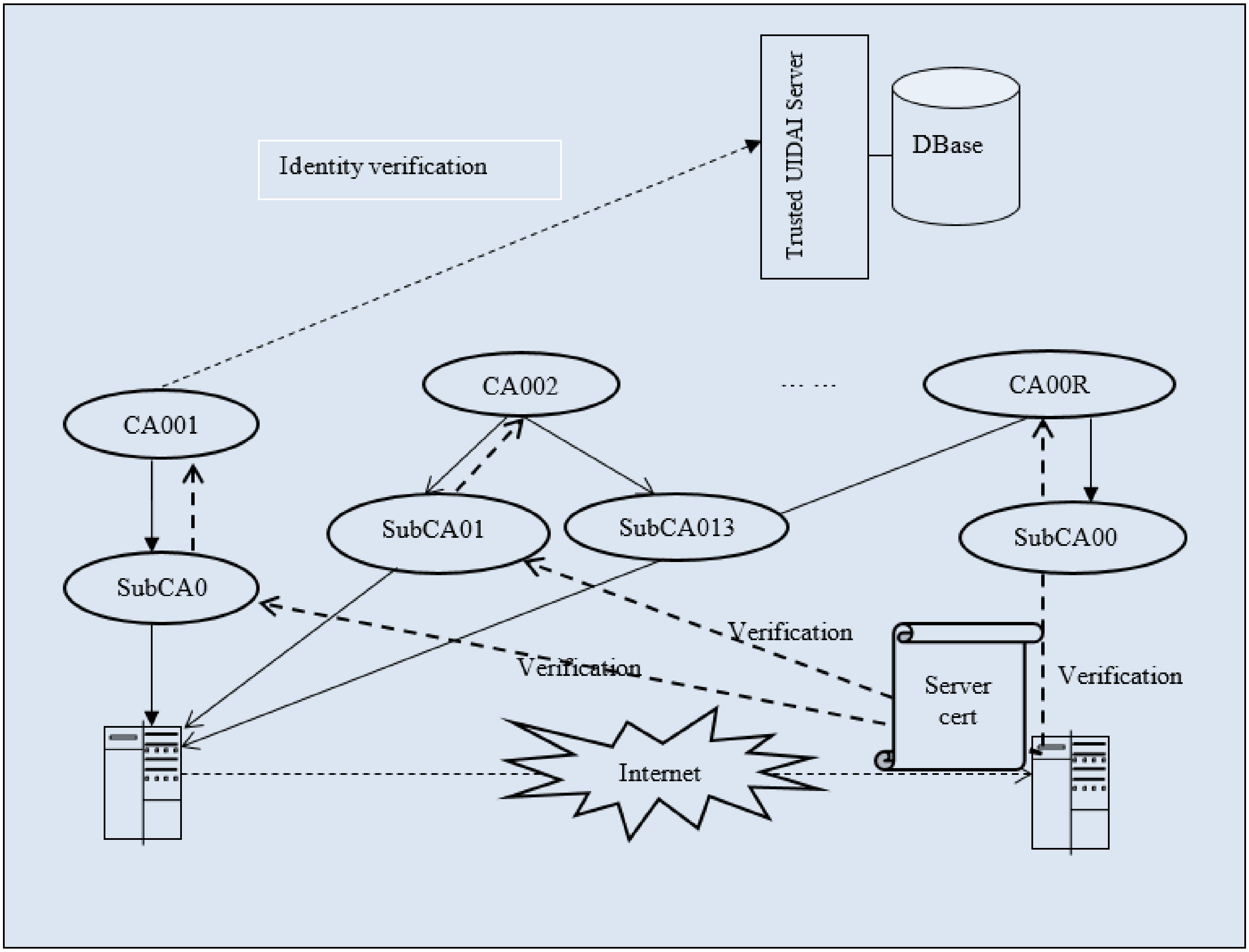
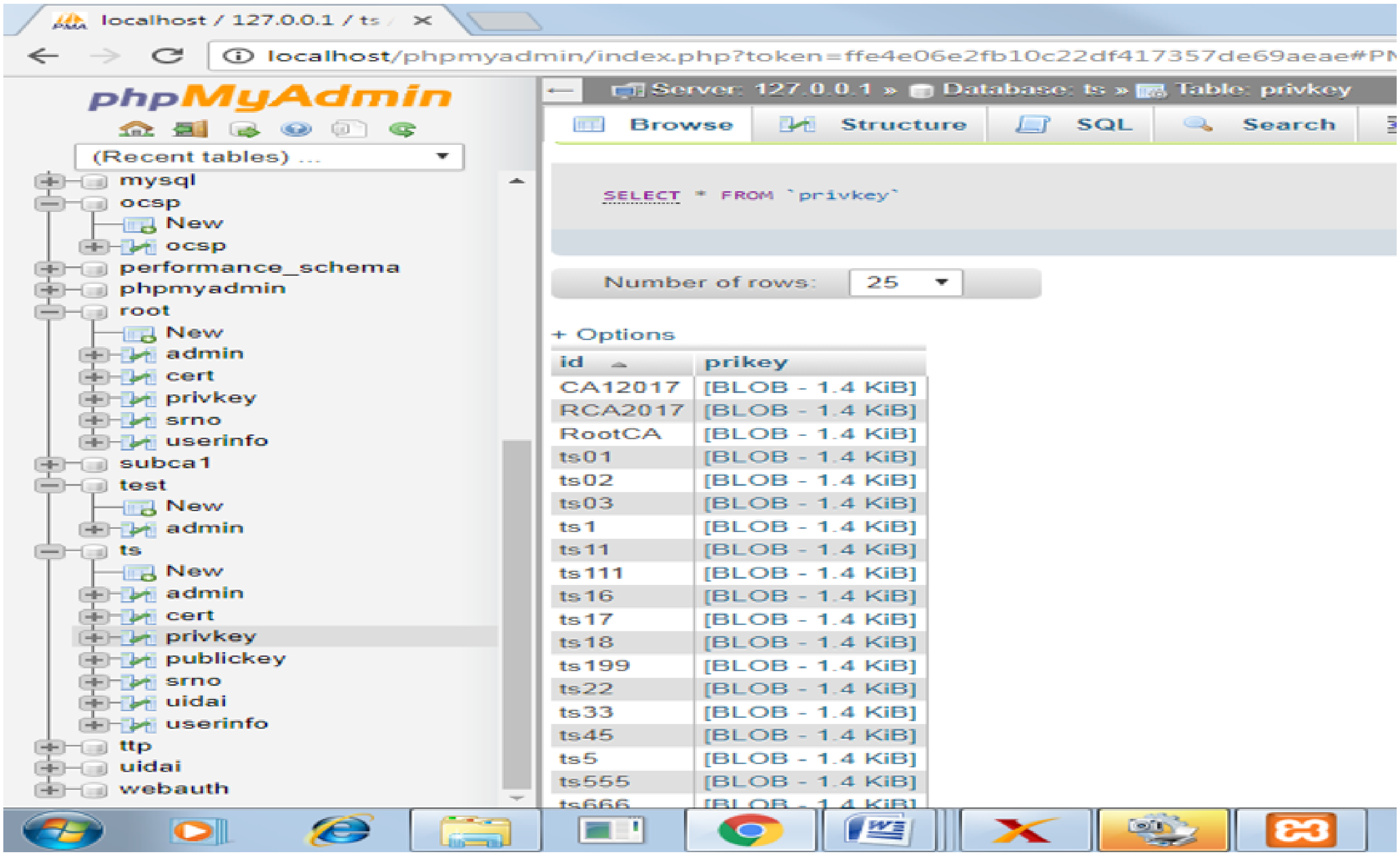


Figure 11

Proposed approach for Certificate Verification

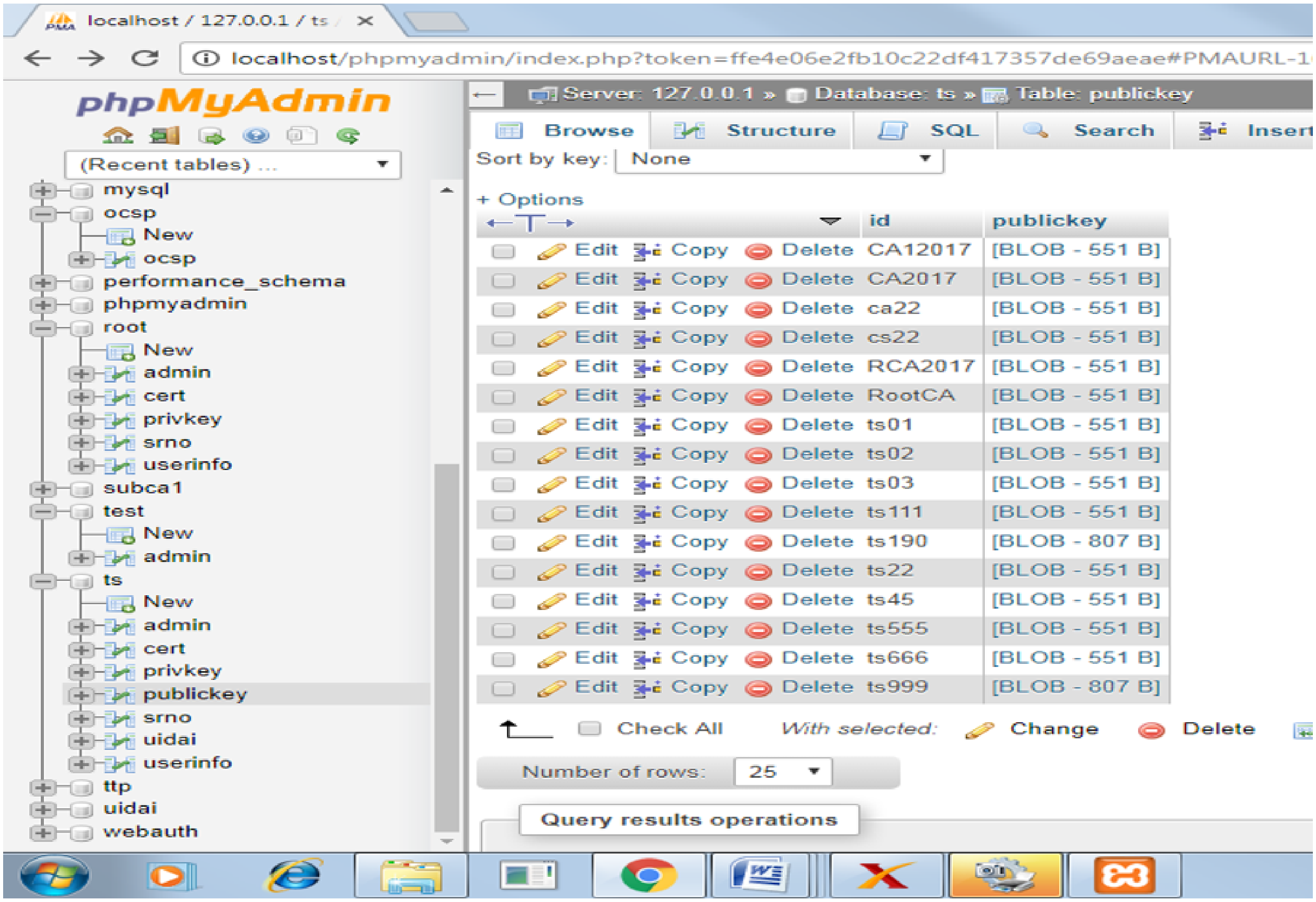


```
SELECT * FROM `privkey`
```

id	prikey
CA12017	[BLOB - 1.4 KiB]
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]

Figure 12

Private Key storage in Blob Form



localhost/phpmyadmin/index.php?token=ffe4e06e2fb10c22df417357de69aeae#PMAURL-1

Server: 127.0.0.1 » Database: ts » Table: publickey

Sort by key: None

←T→				id	publickey
□	Edit	Copy	Delete	CA12017	[BLOB - 551 B]
□	Edit	Copy	Delete	CA2017	[BLOB - 551 B]
□	Edit	Copy	Delete	ca22	[BLOB - 551 B]
□	Edit	Copy	Delete	cs22	[BLOB - 551 B]
□	Edit	Copy	Delete	RCA2017	[BLOB - 551 B]
□	Edit	Copy	Delete	RootCA	[BLOB - 551 B]
□	Edit	Copy	Delete	ts01	[BLOB - 551 B]
□	Edit	Copy	Delete	ts02	[BLOB - 551 B]
□	Edit	Copy	Delete	ts03	[BLOB - 551 B]
□	Edit	Copy	Delete	ts111	[BLOB - 551 B]
□	Edit	Copy	Delete	ts190	[BLOB - 807 B]
□	Edit	Copy	Delete	ts22	[BLOB - 551 B]
□	Edit	Copy	Delete	ts45	[BLOB - 551 B]
□	Edit	Copy	Delete	ts555	[BLOB - 551 B]
□	Edit	Copy	Delete	ts666	[BLOB - 551 B]
□	Edit	Copy	Delete	ts999	[BLOB - 807 B]

Number of rows: 25

Figure 13

Public Key Directory

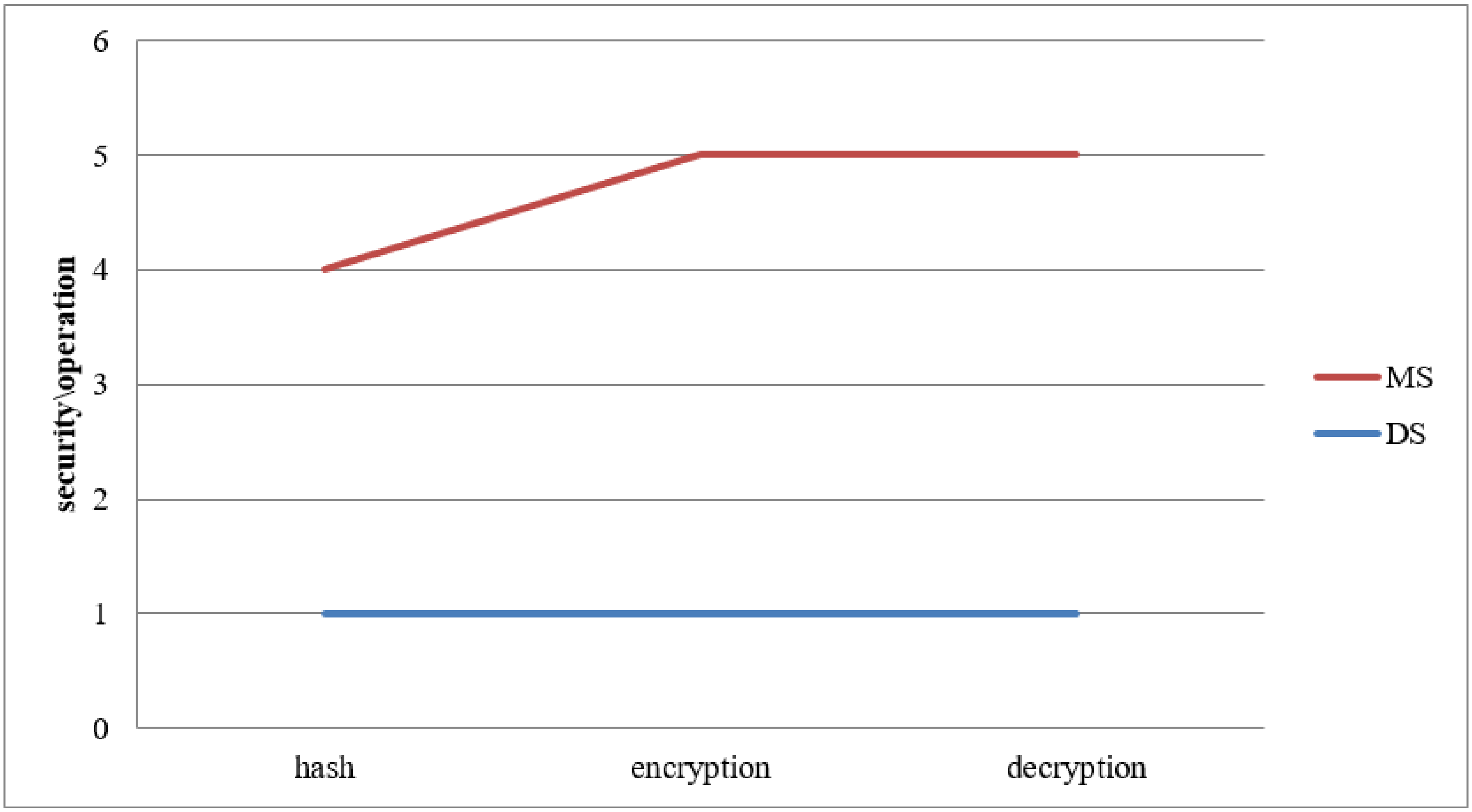


Figure 14

No. of Operation/Security Analysis

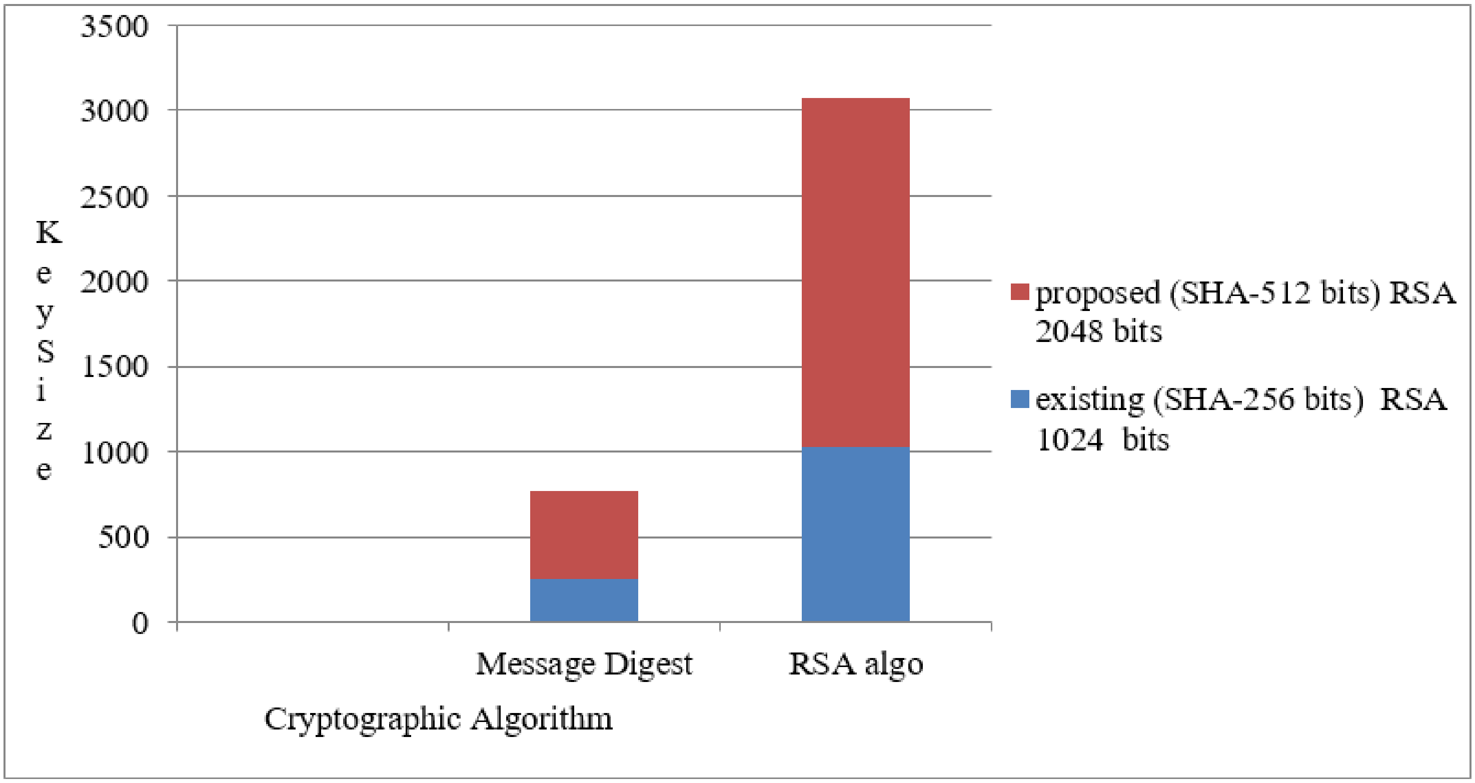


Figure 15

Cryptographic Algorithm Comparison

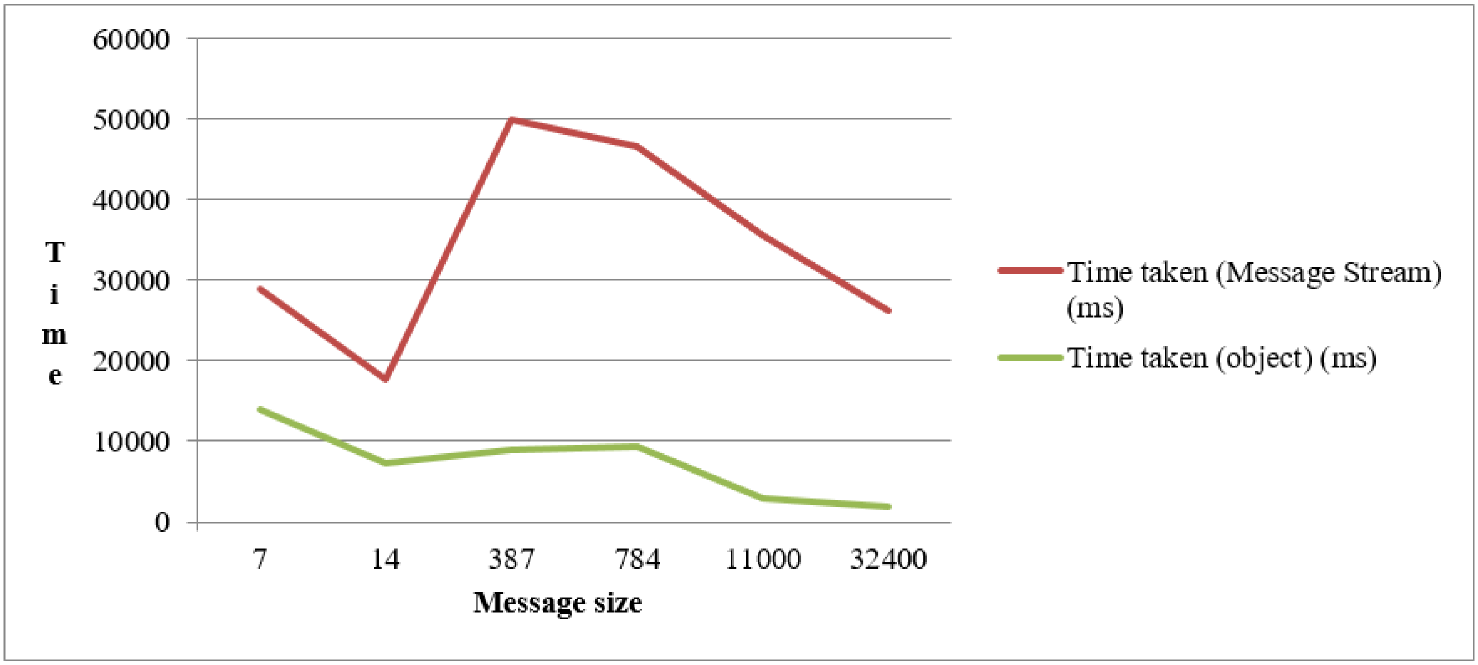


Figure 16

Fast transfer of Information as Compared to Existing Approaches