

**SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING**

**A Project Report**

**On**

**AUTHENTICATION BASED HYBRID CRYPTOSYSTEM**

*Submitted in partial fulfilment of the requirements for the degree of*

**Bachelor of Computer Applications**

*By*

*LINGAMOORTHY V (20BCA0060)*

*GURUPRASATH A (20BCA0118)*

**Under the guidance of**

Prof CHANDRASEGAR.T

**April 2023**

**Signature of Guide:**

**DECLARATION**

I hereby declare that the thesis entitled “AUTHENTICATION BASED HYBRID CRYPTOSYSTEM” submitted by me, for the award of the degree of Specify the name of the degree VIT is a record of bonafide work carried out by me under the supervision of PROF.CHANDRASEGAR.T

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore LINGAMOORTHY V

GURUPRASATH A

Date: Signature of the Candidate

**CERTIFICATE**

This is to certify that the thesis entitled “AUTHENTICATION BASED HYBRID CRYPTOSYSTEM” submitted by LINGA MOORTHY V (20BCA0060) GURUPRASATH A (20BCA0118) School of the Information Technology and Engineering VIT for the degree of the M-Tech Software Engineering is a record of bonafide work carried out under the supervision of PROF.CHANDRASEGAR.T.

The contents of this report have not been submitted and will not be submitted  
either in part or in full, for the award of any other degree or diploma in this institute or  
any other institute or university. The Project report fulfils the requirements and regulations of  
VIT and in my opinion meets the necessary standards for submission.

**Signature of the Guide**  **Signature of the Hod**

**Internal Examiner**  **External Examiner**

**ACKNOWLEDGEMENT**

It is my pleasure to express with deep sense of gratitude to PROF.CHANDRASEGAR.T, Assistant Professor Senior Grade, SITE, Vellore Institute of Technology, for his constant guidance, continual encouragement, understanding more than all, he taught me patience in my endeavour. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Cryptography and Biometric systems.

I would like to express my gratitude to DR.G.VISWANATHAN, Chancellor VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, MR. SANKAR VISWANATHAN, DR. SEKAR VISWANATHAN, MR.G V SELVAM, Vice – Presidents VELLORE INSTITUTE OF TECHNOLOGY, VELLORE, DR. RAMBABU KODALI, Vice – Chancellor, DR. S. NARAYANAN, Pro-Vice Chancellor and Dr. S. Sumathy, Dean, School of Information Technology & Engineering (SITE), for providing with an environment to work in and for his inspiration during the tenure of the course

In jubilant mood I express ingeniously my whole-hearted thanks to Dr.Parimala M, HoD/Professor, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore LINGAMOORTHY V

GURUPRASATH A

Date: Name of the student

**CONTENTS**

**CONTENTS**

**LIST OF FIGURE**

**LIST OF TABLES**

**LIST OF ACRONYMS**

**CHAPTER 1**

**INTRODUCTION**

* 1. INTRODUCTIO
  2. SCOPE

1.3 EXISTING SYSTEM

1.4 OBJECTIVE

1.5 PROBLEM STATEMENT

**CHAPTER 2**

**LITERATURE SURVEY**

2.1 INTRODUCTION

2.2 RELATED WORKS

2.3 FUNCTIONAL REQUIREMENTS

2.4 NON-FUNCTIONAL REQUIREMENT

*iii)*

**CHAPTER 3**

**SYSTEM ANALYSIS**

3.1 PROPOSED SYSTEM

3.2 BLIND SIGNATURE AND ALGORITHM

3.3 LINEAR PKC WITH BLIND SIGNATURE

3.4 UML DIAGRAMS

**CHAPTER 4**

**SOFTWARE DESCRIPTION**

4.1 PROGRAMMING LANGUAGE

4.2 SYSTEM REQUIREMENTS

**CHAPTER 5**

**IMPLEMENTATION**

5.1 SAMPLE CODE

**CHAPTER 6**

**RESULTS**

6.1 SCREENSHOTS

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

7.1 CONCLUSION AND FUTURE WORK

**CHAPTER 8**

8.1 REFERENCES

**LIST OF FIGURES**

1.1 SYMMETRIC ENCRYPTION

1.2 ASYMMETRIC ENCRYPTION

2.1 SYSTEM ARCHITECTURE

3.1. LINEAR PKC ARCHITECTURE

3.2. USE CASE DIAGRAM

3.3. SEQUENCE DIAGRAM

3.4. CLASS DIAGRAM

4.1. JAVA VIRTUAL MACHINE

*v)*

**LIST OF TABLES**

2.1. CHAN-PKC INSTANCES

3.1. LINEAR PKC INSTANCES

6.1. BLIND SIGNATURE TEST CASES AND RESULTS

6.2 COMPARISON OF CHAN-PKC AND LINEAR

**LIST OF ACRONYMS**

PKC PUBLIC KEY CRYPTOSYSTEM

AES ADVANCED ENCRYPTION STANDARD

DES DATA ENCRYPTION STANDARD

CT CIPHER-TEXT

**CHAPTER 1**

##### **Introduction:**

##### CRYPTOGRAPHY:

Cryptography is a technique to achieve confidentiality of messages. . Cryptography can be defined as techniques that cipher data, depending on specific algorithms that make the data unreadable to the human eye unless decrypted by algorithms that are predefined by the sender. Nowadays, however, the privacy of individuals and organizations is provided through cryptography at a high level, making sure that information sent is secure in a way that the authorized receiver can access this information. In order to achieve this level of security, various algorithms and methods have been developed. Billions of people around the globe use cryptography on a daily basis to protect data and information, although most do not know that they are using it. In addition to being extremely useful, it is also considered highly brittle, as cryptographic systems can become compromised due to a single programming or specification error. Cryptography plays a vital and critical role in achieving the primary aims of security goals, such as authentication, integrity, confidentiality, and no-repudiation. Cryptographic algorithms are developed in order to achieve these goals. Cryptography has the important purpose of providing reliable, strong, and robust network and data security.

TYPES:

\* SYMMETRIC CRYPTOGRAPHY

\* ASYMMETRIC CRYPTOGRAPHY

\* HYBRID CRYTOGRAPHY

\* ENSEMBLE CRYPTOGRAPHY

1

SYMMETRIC CRYPTOGRAPHY:

[Single-key or symmetric-key encryption](https://searchsecurity.techtarget.com/feature/Cryptography-basics-Symmetric-key-encryption-algorithms) algorithms create a fixed length of bits known as a [block cipher](https://searchsecurity.techtarget.com/definition/block-cipher) with a secret key that the creator/sender uses to encipher data (encryption) and the receiver uses to decipher it. Types of symmetric-key cryptography include the [Advanced Encryption Standard](https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard) (AES).By using symmetric encryption algorithms, data is converted to a form that cannot be understood by anyone who does not possess the secret key to decrypt it. Once the intended recipient who possesses the key has the message, the algorithm reverses its action so that the message is returned to its original and understandable form. The secret key that the sender and recipient both could be a specific password or code or it can be random string of letters or numbers that have been generated by a secure random number generator.

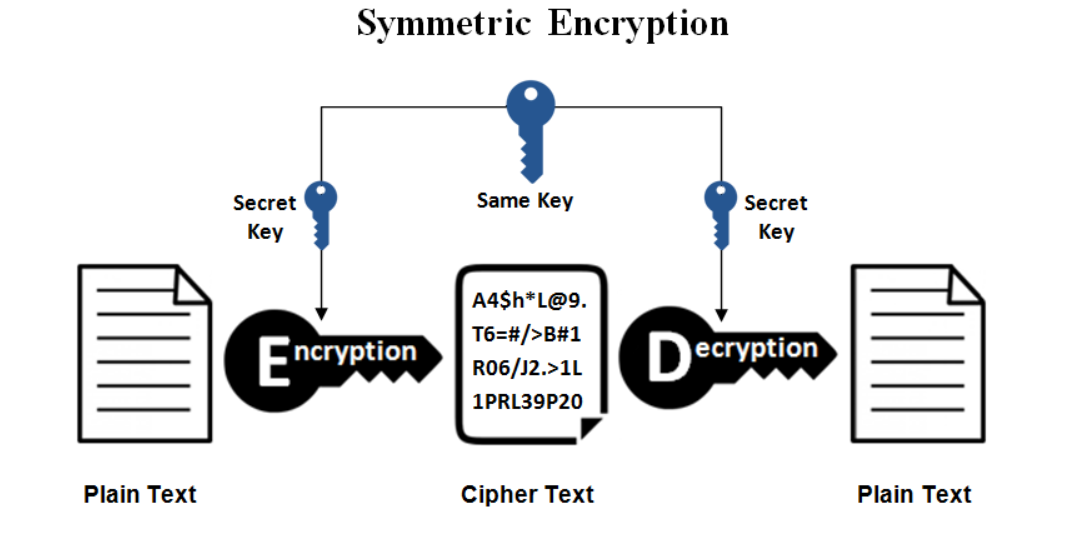


Fig 1.1

2

ASYMMETRIC CRYPTOGRAPHY:

Public-key cryptography or asymmetric cryptography, is a cryptographic system that uses pairs of keys public keys, which may be disseminated widely and private keys, which are known only to the owner. The generation of such keys depends on cryptographic algorithms based on mathematical problems to produce one-way functions. Effective security only requires keeping the private key private; the public key can be openly distributed without compromising security. In such a system, any person can encrypt a message using the receiver’s public key but that encrypted message can only be decrypted with the receiver’s private key. This allows, for instance, a server to generate a cryptographic key intended for symmetric-key cryptography, then use a client’s openly-shared public key to encrypt that newly-generated symmetric key.

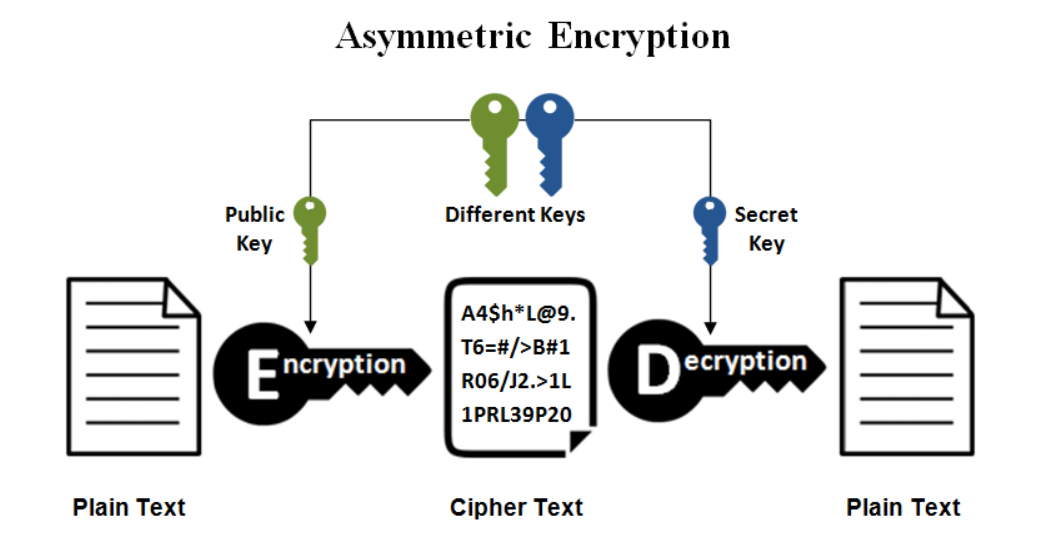


Fig 1.2

3

HYBRID CRYPTOGRAPHY:

Hybrid cryptography is a system which is a combination of symmetric cryptography and the asymmetric cryptography. It uses both the keys of the cryptography. Public-key cryptosystems are convenient in that they do not require the sender and receiver to share a common secret in order to communicate securely (among other useful properties). However, they often rely on complicated mathematical computations and are thus generally much more inefficient than comparable symmetric-key cryptosystems. In many applications, the high cost of encrypting long messages in a public-key cryptosystem can be prohibitive. This is addressed by hybrid systems by using a combination of both.

ENSEMBLE CRPTOGRAPHY:

Ensemble cryptography is a system which is a combination of hybrid cryptography to hybrid and asymmetric cryptography to asymmetric cryptography and symmetric cryptography to symmetric cryptography. Here both the private key and the public key are used for both the encryption and the decryption process.

* 1. SCOPE OF PROJECT:

Our project's goal is to create a new algorithm using the Linear RSA and RSA to improve the security by using minimal bit with less time consumption ,Blind signatures can also be used to provide unlink-ability which prevents the signer from linking the blinded message it signs to a later un-blinded version that it may be called upon to verify. In this case, the signer’s response is first un-blinded prior to verification in such a way that the signature remains valid for the un-blinded message. To perform such a signature, the message is first “blinded”, typically by combining it in some way with a random blinding factor. The blinded message is passed to a signer, who then signs it using a standard signing algorithm. The resulting message, along with the blinding factor, can be later verified against the signer’s public key.

1.3 EXISTING SYSTEM:

In the standard RSA public key cryptosystem contains three main phases, such as key generation, encryption, and decryption. 1. In a key generation, asymmetric keys are produced, such as public keys (e, N) and private key (d, N). An RSA digital signature key pair consists of an RSA private key, which is used to compute a digital signature, and an RSA public key, which is used to verify a digital signature. An RSA key pair used for digital signatures shall only be used for one digital signature scheme. In addition, an RSA digital signature key pair shall not be used for other purposes (e.g., key establishment). In order to provide security for the digital signature process, the two integer p and q, and the private key exponent d shall be kept secret. The modulus n and the public key exponent e may be made known to anyone. The security strength associated with the RSA digital signature process is no greater than the minimum of the security strength associated with the bit length of the modulus and the security strength of the hash function that is employed.

* 1. OBJECTIVE:

To create and ensure the various phases of hybrid cryptosystem. Such as

* key generation
* encryption
* decryption
* sign generation
* sign verification.

1.5 PROBLEM STATEMENT:

* The RSA apply one level of encryption and decryption, whereas proposed HYBRID CRYPTOSYSTEM scheme generates three levels of encryption and decryption.
* Obtain the Blind factor and Blind message.
* Sign generation and sharing the sign.
* Verifying the signature by verifying blind message and message.
* The cost of establishing and utilizing certification authorities, repositories, and other important services, as well as assuring quality in the performance of their functions.

NOVELTY:

* Implementing Blind Signature Technique in HYBRID CRYPTOSYSTEM.
* Private and public key generation.
* Sign generation.
* Sign verification.

**CHAPTER 2**

**LITERATURE SURVEY:**

2.1 LITERATURE SURVEYS ON PROJECT:

Related works:

TITLE 1:Hybrid IT architecture by gene-based cryptomata (HITAGC) for lightweight security services

Author: Viswanathan P.

Abstract: In the computing world, the digital transformation of data grows exponentially with every year. However, these situations are predominantly tackled by leading enterprises through by adopting a hybrid IT model. This model effectively supports the organization through strategic design to provide standard delivery to customers from independent multi-sourced entities. An experimental design method is applied using hybrid IT architecture with gene-based cryptomata (HITAGC) which is well suited to real-time cloud environments to store and retrieve the cloud space data efficiently. The CSC primarily relies on the service of CSP to keep their confidential data, and this system ultimately uses the PKC crypts. To address this security demand, we propose two different schemes such as HITAGC-PKC and hybrid HITAGC. The first model is designed to generate robust asymmetric keys for the Internet of Things, and the other hybrid HITAGC is for efficient sharing of big data over the cloud in a secure manner. The performance of these crypts is compared with the traditional systems like standard RSA, ESRKGS, SED2, EDCon, and AES. Keywords Symmetric encryption · Hybrid model Asymmetric cryptography · Big data · Internet of Things · Lightweight.

Algorithm of HITAGC:

The proposed scheme-I involves the usage of secret language. Firstly, our scheme is given with key generation which contains parameters (A, G, D, C) in which the C is the common moduli, A is the public key, and (G, D) are the private key. Moreover, these triplet keys (A, G, D) can be used in different forms, in which the user can obtain either single key or dual key for encryption. For instance, single key and dual key combinations are (A, C), (G, C), (D, C) and (G, D, C),(A, D,C), (A, G,C), respectively.

The strength of our scheme-I is based on the selection of a secret language for the server secret variable H1, one-to-one substitution H S 1 ,the nearest prime on I, and the secret variables on T . The component H+1 ∗ H S 1 is computed for the generation of a secret variable T.

STEPS:

1. Define a secret language H = {x\*n (0+1), where 2Nm ≤ x\*n ≤ N2 m (Nm + 1)/2}

2. Compute H1 = {x1, x2, x3, x\*n} which is the server unique secret variable of n clients and considered to be the genotypes of number theory.

3. Using substitution, define H S 1 = {s (xn+1) = x\*n x1, s (xn+2) = x\*n x2s (x\*n+ n) = x\*n x\*n}

4. Compute (H+1 ∗ H S) where H+1 is randomly selected from the set H1.

5. Convert the elements of H+1 to its nearest prime value on I. Using hypothesis zero [34], the prime gene elements are generated on the set I.

6. Compute the secret variable, T = (H+∗ HS1 – I)

7. Compute the public key,

A = (random integer ∗ T) + H+1

8. Compute the private key,

G = (random integer ∗ T) + HS

9. Compute the common moduli,

C = (AG – I)/T

10. Use multiplicative inverse, compute the secret D as follows,

D = I −1mod C

Encryption:

Encryption: Cipher text, CT = M ∗ A mod C.

Decryption:

Decryption: Plain Text, PT = CT ∗ G ∗ D mod C.

Merits:

It provides confidentiality and authentication

Time complexity and time consumed is less.

TITLE 2: An Enhanced and Secured RSA Key Generation Scheme (ESRKGS)

Author: M. Thangavel, P. Varalakshmi,

ABSTRACT: Public-key cryptography can be claimed as the greatest and an excellent revolution in the field of cryptography. A public-key cryptosystem is used for both confidentiality and authentication. One such public-key cryptosystem is the RSA cryptosystem. In this paper, a modified and an enhanced scheme based on RSA public-key cryptosystem is developed. The proposed algorithm makes use of four large prime numbers which increases the complexity of the system as compared to traditional RSA algorithm which is based on only two large prime numbers. In the proposed Enhanced and Secured RSA Key Generation Scheme (ESRKGS), the public component n is the product of two large prime numbers but the values of Encryption € and Decryption (D) keys are based on the product of four large prime numbers (N) making the system highly secured. With the existing factorization techniques, it is possible only to find the primes p and q. The knowledge of n alone is not sufficient to find E and D as they are based on N. The time required for cryptanalysis of ESRKGS is higher than traditional RSA cryptosystem. Thus the system is highly secure and not easily breakable. A comparison is done between the traditional RSA Scheme. A recent RSA modified scheme and our scheme to show that the proposed technique is efficient.

The proposed ESRKGS key generation involves the usage of four prime numbers. The value of E, D depends on the value of N, which is the product of 4 prime numbers. The computation of E is also not direct. The values e1, e2 are needed to find the value of E1 thus increasing the time taken to attack the system. Only the value of n is kept as public and private component. Thus the attacker with the knowledge of n cannot determine all the primes which are the basis for finding the value of N and subsequently D. The parameter E1 also increases the complexity of the system. For security purposes, the bit length of all the primes chosen is of same length as in case of traditional RSA.

Algorithm of ESRKGS:

Key Generation:

**INPUT:**

Four prime numbers p, q, r, s

**OUTPUT:**

Public Key components :{ E, n}

Private Key components :{ D, n}

**PROCEDURE:**

n = p\*q

m = r\*s

N = n\*m

Compute Euler phi value of n and m

Φ (n) = (p-1)\*(q-1)

Φ (m) = (r-1)\*(s-1)

Compute Euler phi value of N

Φ (N) = Φ (n)\* Φ (m)

Find a random number e1

Gcd (e1, Φ (n)) = 1

Find a random number e2

Gcd (e2, Φ (m)) = 1

Find a random number E

Gcd (E, Φ (N)\*E1) = 1

OUTPUT:

Cipher text, C

OUTPUT:

Decrypted Plain text, P

**Encryption:**

**INPUT:**

Plain text message, M (<n)

Public key components {E, n}

**OUTPUT**:

Cipher text, C

**Procedure:**

C- ME mod n

**Decryption:**

**INPUT:**

Cipher text message, M (<n)

Private keys components {E, n}

**OUTPUT:**

Decrypted Plain text, P

**PROCEDURE:**  
P - CD mod n

MERITS:

This algorithm is highly secure and not easily breakable as compared to RSA and the compared modified RSA algorithm.

TITLE 3:An efficient public key secure scheme for cloud and IOT security

Author: Senthil Kumar Mohan.

Abstract:

According to the National Institute of Standard and Technology (NIST), the security level of RSA is safe when it is N-bit modulus ≥2048 bits. Because of this, the processing time to generate asymmetric keys also increases. Taking this into account, an efficient and non-shareable Public Key Exponent Secure Scheme (ENPKESS) is proposed by using a non-linear Diophantine equation to have high security against side-channel attacks like timing attacks. This scheme has three-stage of encryption and two-stage of decryption, whereas other schemes like ESR and RSA has one level in encryption and decryption. Due to this, extraction of the secret key is extremely hard to determine from our public exponents α, R, N. Our methodology is well suited for secure cloud computing environments used in the Internet of Things (IoT). Here we have also applied the Knapsack method to encrypt our ENPKESS keys to enrich high security in cloud systems. We show a strong performance evaluation on standard RSA, Enhanced and Secured RSA Key Generation Scheme (ESRKGS), and ENPKESS on its key generation, encryption and decryption by varying the N-bit moduli size up to 10K bits. From the overall result, ENPKESS consumes 89% of standard RSA and 27% of ESRKGS.

Merits:

* It is more robust.
* It is less susceptible to third-party security breach attempts.

TITLE 4:Intelligent cryptography approach for secure distributed big data storage in cloud computing

Author: Yibin Li.

Abstract:

Implementing cloud computing empowers numerous paths for Web-based service offerings to meet diverse needs. However, the data security and privacy has become a critical issue that restricts many cloud applications. One of the major concerns in security and privacy is caused by the fact that cloud operators have chances to reach the sensitive data. This concern dramatically increases users’ anxiety and reduces the adoptability of cloud computing in many fields, such as the financial industry and governmental agencies. This paper focuses on this issue and proposes an intelligent cryptography approach, by which the cloud service operators cannot directly reach partial data. The proposed approach divides the file and separately stores the data in the distributed cloud servers. An alternative approach is designed to determine whether the data packets need a split in order to shorten the operation time. The proposed scheme is entitled Security-Aware Efficient Distributed Storage (SA-EDS) model, which is mainly supported by our proposed algorithms, including Alternative Data Distribution (AD2) Algorithm, Secure Efficient Data Distributions (SED2) Algorithm and Efficient Data Conflation (EDCon) Algorithm. Our experimental evaluations have assessed both security and efficiency performances and the experimental results depict that our approach can effectively defend main threats from clouds and requires with an acceptable computation time.

Merits:

* It is more safe and secure.

TITLE 5: An Enhanced and Secured RSA Key Generation Scheme (ESRKGS**)**

Author: Thangavel M.

Abstract:

Public-key cryptography can be claimed as the greatest and an excellent revolution in the field of cryptography. A public-key cryptosystem is used for both confidentiality and authentication. One such public-key cryptosystem is the RSA cryptosystem. In this paper, a modified and an enhanced scheme based on RSA public-key cryptosystem is developed. The proposed algorithm makes use of four large prime numbers which increases the complexity of the system as compared to traditional RSA algorithm which is based on only two large prime numbers. In the proposed Enhanced and Secured RSA Key Generation Scheme (ESRKGS), the public component n is the product of two large prime numbers but the values of Encryption € and Decryption (D) keys are based on the product of four large prime numbers (N) making the system highly secured. With the existing factorization techniques, it is possible only to find the primes p and q. The knowledge of n alone is not sufficient to find E and D as they are based on N. The time required for cryptanalysis of ESRKGS is higher than traditional RSA cryptosystem. Thus, the system is highly secure and not easily breakable. A comparison is done between the traditional RSA scheme, a recent RSA modified scheme and our scheme to show that the proposed technique is efficient.

Merits:

* It provides confidentiality and authentication.
* Time complexity is also reduced

TITLE 6: Memory efficient multi key (MEMK) generation scheme for secure transportation of sensitive data over cloud and IOT devices.

Author: Himanshu.

Abstract: A new variant of RSA has been proposed called Memory Efficient Multi Key (MEMK) generation scheme. For sensitive data, our scheme will aid in exchanging the information between cloud to IOT and IOT to IOT devices. When cryptography belongs to the asymmetric type, then it has public and private keys. For memory efficiency, our scheme reuses the RSA scheme with a Diophantine form of the nonlinear equation. Moreover, our scheme performance comparatively performs well and this mainly due to the use of RSA public key alone. Due to this, our MEMK does not require multiplicative inverse function or Extended Euclid’s algorithm. Finally, we have made an experimental result on various phases of MEMK PKC such as key generation, encryption, and decryption by varying the N-bit modulo bits from 1K to 10K.

Merits:

\* It helps in exchange of information between IOT to cloud and between IOT to IOT Devices.

\* It comparatively performs well and this mainly due to the use of RSA public key alone.

Demerits:

\* But it takes more time to complete and time complexity is high.

TITLE 7: Efficient Cancellable Biometric Key Generation Scheme for Cryptography

Author: Sunil V K.

Abstract: This paper puts forth a fresh methodology for the secure storage of fingerprint template by generating Secured Feature Matrix and keys for cryptographic techniques applied for data Encryption or Decryption with the aid of cancellable biometric features. Conventional techniques depend on biometric features like face, fingerprint, hand geometry, iris, signature, keystroke, voice and the like for the extraction of key information. If a Biometric Key is missing or stolen, it is lost perpetually and possibly for every application where the biometric is utilized, since a biometric is permanently linked with a user and cannot be altered. In this paper we propose a technique to produce cancellable key from fingerprint so as to surmount these problems. The flexibility and dependability of cryptography is enhanced with the utilization of cancellable biometric features. There are several biometric systems in existence that deal with cryptography, but the proposed cancellable biometric system introduces a novel method to generate Cryptographic Key. We have as well discussed about the Security analysis of the projected Cancellable Biometric System.

Merits:

\* The proposed cancellable biometric system introduces a novel method to generate Cryptographic Key.

\* The flexibility and dependability of cryptography is enhanced with the utilization of cancellable biometric features.

TITLE 8: Physical Layer Cryptographic Key Generation by Exploiting PMD of an Optical fiber Link

Author: Imam UZ Zaman.

Abstract: We present a symmetric physical layer based secret key generation scheme for Point-to-Point Optical Link (PPOL) communication by exploiting Polarization Mode Dispersion (PMD) as a random and inimitable channel characteristic. The randomness and security strength of generated cryptographic keys based on PMD is significantly high. In this paper, we present that random modulation of a probe signal caused by PMD in a high-speed data communication network (40 Gb/s and 60 Gb/s) is reciprocal with average Pearson correlation coefficient of 0.862, despite the presence of optical nonlinearities, dispersion, and noise in the system. 128-bit symmetric cryptographic key has been successfully generated using the proposed scheme. Moreover, PMD-based encryption keys passed the National Institute of Standards and Technology (NIST) tests. We have shown through simulations with a 50 km link that, with optimal key generation settings, symmetric keys can be generated with high randomness (high P-values for NIST randomness tests) and with sufficient generation rates (>50%). Furthermore, we considered an attack model of a non-invasive adversary intercepting at 10 km into the link and found that the generated keys have high average key bit mismatch rates (>40%).

Merits:

\*It helps in Point-to-Point Optical Link communication by exploiting Polarization Mode Dispersion as a random and inimitable channel characteristic.

\* The randomness and security strength of generated cryptographic keys based on PMD is significantly high

TITLE 9: Security of Blind Signatures under Aborts

Author: Marc Fischlin and Dominique Schroder.

Abstract: We explore the security of blind signatures under aborts where the user or the signer may stop the interactive signature issue protocol prematurely. Several works on blind signatures discuss security only in regard of completed executions and usually do not impose strong security requirements in case of aborts. One of the exceptions is the paper of Camenisch, Neven and shelat (Eurocrypt 2007) where the notion of selective-failure blindness has been introduced. Roughly speaking, selective-failure blindness says that blindness should also hold in case the signer is able to learn that some executions have aborted. Here we augment the work of Camenisch et al. by showing how to turn every secure blind signature scheme into a selective-failure blind signature scheme. Our transformation only requires an additional computation of a commitment and therefore adds only a negligible overhead. We also study the case of multiple executions and notions of selective-failure blindness in this setting. We then discuss the case of user aborts and unforgetability under such aborts. We show that every three-move blind signature scheme remains unforgeable under such user aborts. Together with our transformation for selective failure blindness we thus obtain an easy solution to ensure security under aborts of either party and which is applicable for example to the schemes of Pointcheval and Stern (Journal of Cryptology, 2000). We finally revisit the construction of Camenisch et al. for simulatable adaptive oblivious transfer protocols, starting from selective-failure blind signatures where each message only has one valid signature (uniqueness). While our transformation to achieve selective-failure blindness does not preserve uniqueness, it can still be combined with a modified version of their protocol.

Merits:

* To provide Authenticity, Integrity and Non-repudiation to electronic documents.

2.2 EXISTING SYSTEM:

CHAN-PKC:

The CHAN-PKC scheme outflow from the traditional method of sharing the keys and produces the public key as (α, Re, N). This scheme of KGS takes (R, p, q) plus RSA public key as input and produces valid private key components (e, 2Yl, N) as output. The performance measure of this scheme assessed in standings of key generation, encryption and decryption by varying its input bits. Here, the bit length of RSA public key e lies in half of its N-bit moduli size. CHAN-PKC scheme works by solving the Pell’s essential form, α + Re2 + 2RYle ≡ 1 mod ∅ (N). The chief merit of our system is that it does not need Extended Euclid’s algorithm as like RSA-PKC. This merit has shown in the below CHAN-PKC key generation scheme. The proposed scheme applies two levels of encryption to create the cipher-texts CT1, and CT2 using the public key (α, Re, N). Using the private key (e, 2Yl, N) three levels of decryption is applied which produces a high level of security and confidentiality. The existing cryptographic strength depends on the factorization complexity. The guessed private key ′d′ has made through the factors of common modulus ′N′ and public key ′e′. However, our proposed scheme strength depends on the complexity of RSA (e, ∅ (N)) parameters and Pell’s coordinates bit length. A CHAN-PKC scheme based on improved RSA public key cryptography with Diophantine equation to have the three stage of decryption for high security.

OBJECTIVE:

The objective of this project is to propose blind signature scheme and to fulfil the requirements of blind signature scheme like correctness, blindness, confidentiality. It is also show the computation cost of proposed scheme. It is intend to improve the performance of voting system or other applications by applying proposed blind signature scheme.

**CHAN-PKC:**

Encryption:

**INPUT:**

The message M and public key (Re, N, α)

**OUTPUT:**

CT1, CT2

Here the unique message M is encrypted into two cipher forms as shown below.

CT1 = M^α mod N

CT2=M^Re mod N

**Decryption:**

**INPUT:**

The cipher-texts, CT1, CT2 and private key (e, 2Y, N)

**OUTPUT:**

M

Here the cipher-texts are decrypted into original message as shown

PT = CT1. CT2^e mod (N).CT2^2Y mod (N)

Numerical instance of CHAN–PKC:

**Key Generation:**

Input: (R, p, q): (19, 673, 937)

Output: Public Key, (α, Re, N): (309048, 275881, 630601) & Private Key: (e, 2Yl, N).

At first, a secret non-square positive integer, R = 19, and randomly selected primes p = 673, q = 937 have taken as inputs. Hence the base coordinates (X0, Y0) of the Diophantine equation stands (170, 39).

The RSA components have shown below:

1. Euler totient function, ∅ (N) = (p – 1). (q− 1) = 672 ∗ 936 = 628992
2. The common modulus, N = p. q = 673 ∗ 937 = 630601

1. Select RSA public key e, satisfying ∅ð Þ N 2 < ∅ (N) and GCD (e, ∅ (N)) = = 1

That is 314496 < e < 628992 and GCD (e, ∅ (N)) = = 1. Hence, the public key of RSA ′e′ has chosen as 444883. For instance, Pell’s co-ordinate Yl has selected as 39. Now the Pell’s public key (α, Re) are computed.

= (170) 2 mod 628992− (19\*(39+ 444883) 2 mod 628992

= 28900 + (-348844 mod 628992) = 309048

Re = Re mod ∅N

= 8452777 mod 628992

= 275881

Encryption:

Input: (R, p, q): (19, 673, 937)

Output: Public Key, (α, Re, N): (309048, 275881, 630601) & Private Key: (e, 2Yl, N).

At first, a secret non-square positive integer, R = 19, and randomly selected primes p = 673, q = 937 have taken as inputs. Hence the base coordinates (X0, Y0) of the Diophantine equation stands (170, 39).

Input: Public Key, (α, Re, N): (309048, 275881, 630601) & a message M = 8.

Output: Cipher-texts, (CT1, CT2): (571376, 331108)

Now, the unique message M has ciphered into two altered forms such as {CT1, CT2}

CT1 = Mα mod N = 8309048 mod 630601 = 571376

CT2 = M Re mod N = 8275881 mod 630601 = 331108

Decryption:

Input: Private Key, (e, 2^Yl, N): (444883, 78, 630601) & the cipher texts (CT1, CT2): (571376, 331108).

Output: Decipher text, (DT):8

Now, the cipher-texts are transformed back into the message.

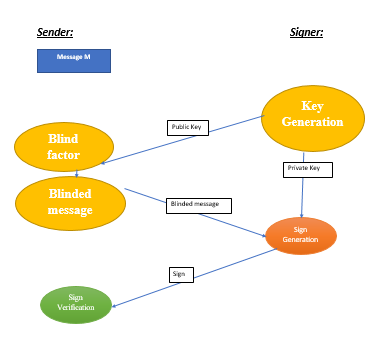
Decipher Text = IDT1IDT2IDT3 = CT1:CT2 e: CT2 2^Yl mod ∅N mod N

= 571376\*331108444883\*3311082\*39 mod 628992mod 630601

= 571376\*566827\*256739 mod 630601

= 8 (Original message)

**PROPOSED METHODOLOGY AND ARCHITECTURE :**



2.2 FUNCTIONAL REQUIREMENTS

HARDWARE REQUIREMENTS:

RAM: 4GB and Higher

• Processor: Intel i3 and above

• Hard Disk: 500GB: Minimum

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

2.3 NON FUNCTIONAL REQUIREMENTS

SOFTWARE REQUIREMENTS

• OS: Windows or Linux

• Net-beans IDE

• JDK package.

• Setup tools and pip to be installed for 3.6 and above

• Language: Java

**CHAPTER 3**

**SYSTEM ANALYSIS**

3.1 PROPOSED SYSTEM:

Blind signature:

Blind Signature is a form of digital signature in which the content of a message is disguised (blinded) before it is signed. The resulting blind signature can be publicly verified against the original, un-blinded message in the manner of a regular digital signature. Blind signatures are typically employed in privacy-related protocols where the signer and message author are different parties. Examples include cryptographic election systems and digital cash schemes.

Algorithm:

\* User have the Message (m).

\* Signer have the Private and Public key.

1 User makes a Sign Request to the Signer.

1. Signer shares his public key with the user.
2. User computes Blind factor.
3. User with the help of Blind factor computes Blind message.
4. User shares the Blind message to the Signer.
5. Signer generates the Sign.
6. User verifies the Sign.

NOVELTY:

1. Implementing Blind Signature Technique in CHAN-PKC.
2. Private and public key generation.
3. Sign generation.
4. Sign verification.

OBJECTIVE:

The objective of this project is to propose blind signature scheme and to fulfil the requirements of blind signature scheme like correctness, blindness, confidentiality. It is also show the computation cost of proposed scheme. It is intend to improve the performance of voting system or other applications by applying proposed blind signature scheme and the main advantage of PKC over symmetric key cryptography is that PKC avoids the pre-distribution of the private key and the symmetric key cryptography has only one key to decrypt the message thus PKC has more advantageous than symmetric key cryptography blind signature scheme and to fulfil the requirements of blind signature scheme like correctness, blindness, confidentiality. It is also show the computation cost of proposed scheme. It is intend to improve the performance of voting system or other applications by applying proposed blind signature scheme.

ADVANTAGES OF BLIND SIGNATURE:

Authentication: The messages may often include information about the entity sending a message, that information may not be accurate. Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. The importance of high confidence in sender authenticity is especially obvious in a financial context.

Integrity: The sender and receiver of a message may have a need for confidence that the message has not been altered during transmission. Although encryption hides the contents of a message, it may be possible to change an encrypted message without understanding it. However, if a message is digitally signed, any change in the message will invalidate the signature. Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions.

BASIC REQUIREMENTS:

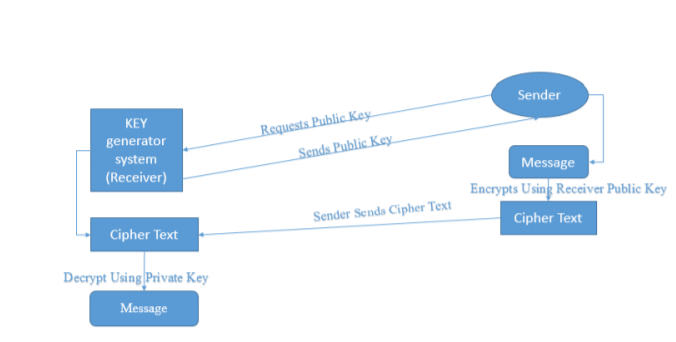
Private Key: The private key is one which is accessible only to the signer. It is used to generate the digital signature which is then attached to the message.

Public Key: The public key is made available to all those who receive the signed messages from the sender. It is used for verification of the received message.

**LINEAR-PKC:**

Linear RSA is a public key cryptography. PKC is a system where the sender, who send a message, to the receiver, by encrypting the message using the public key. This process is called encryption. And after the receiver receives the message from the sender the encrypted message is converted to original plain text using a private key. This process is called decryption. The public and private keys are generated by the algorithm. The main advantage of PKC over symmetric key cryptography is that PKC avoids the pre-distribution of the private key and the symmetric key cryptography has only one key to decrypt the message thus PKC has more advantageous than symmetric key cryptography.

Architecture:



Linear-PKC algorithm:

Key Generation:

Input: p, q, R Output: k, c, L, b 1

* 1. Select three positive integers let’s say (p, q, R) such that R is a prime number and
  2. Defining the variables A, b, c as A=p\*q-R; b=R2A + p; c=RA + q
  3. Computation: b\*c-R= A (R3A 2 +p\*R + qR2 + 1)
  4. Defining: k = (b\*c – R)/A which implies k = R3A 2 +p\*R + qR2 + 1
  5. There is a unique L such that R\*L = 1(mod k) R\*k (By construction P does not divide n) that is GCD (P, k) = 1.

Thus the linear public and private keys are: Public key PU = (c, k) PR = L, b, k

**Encryption and Decryption:**

1. Encryption: Cipher Text CT= (m\*c) mod k where, m is a message.
2. Decryption: Decipher Text DT= (L\*b\*CT) mod k

Numerical instances of Linear-PKC:

Let the three positive number p, q, R be p=13 q=5 and R=13.

Now computing the variable A, b, c we have A = (13\*5)-13; b=52+13; c=52+5.

Defining variable k= ((65\*57)-13)/52.

Now, after the computing and defining variables we need to select a unique L such that R\*L=1(mod k) So by this we get L value as 11 now let’s assume that the message that has to be sent by the sender is m=12 and this message has to be encrypted by using Linear RSA, so by using the above algorithm the encrypted form of the message is that is cipher text is CT= (m\*c) mod k that is CT=(12\*57)%71.

Hence the cipher text is 30. This 12 message is encrypted using the public key and after receiver receives the message the receiver decrypts the message using the private key and the decrypted form of message is DT (decryption text)=(L\*b\*CT)%k that is (11\*65\*30)%71.

Hence, after decryption the retrieved message m is 12 therefore the original message and the decrypted message are same. So by using the public and private key, this algorithm solves the challenges faced by the symmetric key cryptography.

3.3 **Linear RSA with Blind Signature:**

Linear RSA is a public key cryptography. PKC is a system where the sender, who send a message, to the receiver, by encrypting the message using the public key. The public and private keys are generated by the algorithm. The main advantage of PKC over symmetric key cryptography is that PKC avoids the pre-distribution of the private key and the symmetric key cryptography has only one key to decrypt the message thus PKC has more advantageous than symmetric key cryptography and integrating the blind signature in the RSA is the new proposed here. Blind Signature is a form of digital signature in which the content of a message is disguised (blinded) before it is signed. The resulting blind signature can be publicly verified against the original, un-blinded message in the manner of a regular digital signature. Here Linear RSA’s parameters are used for the integrating the blind signature. First with the help of public key blind factor and the blind message is being generated. And Blind message is shared to obtain the sign. After the sign is generated it should be verified if it matching with the message, then the sign is being accepted.

BASIC REQUIREMENTS:

Private Key: The private key is one which is accessible only to the signer. It is used to generate the digital signature which is then attached to the message.

Public Key: The public key is made available to all those who receive the signed messages from the sender. It is used for verification of the received message.

Proposed Algorithm for Linear PKC with Blind signature:

* Receiver generate public & private key using LRSA (Q,E,D,N) public(Q,N) private(E,D)
* Share public key with sender(Q,N)
* Select Random integers R such that GCD(R,N=1)
* Compute Blind Factor. BF=R\*Q mod N
* Compute Blind Message. BM=BF\*m mod N.
* Share BM to receiver to obtain sign.
* Sign Generation: S= BM\*E\*D mod N
* Share the sign with Sender.
* Sign verification: S\*Q mod N gives Blind message.
* Insert R^-1 and Q^-1 gives Message.
* If the obtained message is the same original, then the sign is accepted or else reject the sign.

Advantages of this Proposed System:

Authentication: Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user.

Integrity: If a message is digitally signed, any change in the message will invalidate the signature. Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions.

Confidentiality & Non-Repudiation.

Numerical instance of Linear-blind sign:

Key Generation:

* Let the three positive number a, b, P be a=3 b=7 and P=11.
* Compute M = ab-P
* Compute E = M + a = 13
* Compute D = M + b = 17
* Compute N = E\*D-P/M = 21
* Compute Q = P mod inverse N = 2

Sign generation:

* Compute Blind factor = r \* Q mod N = 5 (r is random Prime number)
* Compute Blind message: BF\*m mod N = 19 (m is message=8)
* Blinded message is being sent to the receiver.
* Sign generation = BM\*E\*D mod N = 20

Sign verification:

* Sign verification t1= S\*Q mod N = 19 ( gives Blind message)
* t2 = t1\* r-1 mod N =16 (gives intermediate sign)
* t = t2\*E\*D mod N =8 (gives message)

If sign matches the message, then accept the sign.

Test Cases for Encryption and Decryption:

Encryption: Encryption here gives the intermediate sign. So if it matches then the sign obtained is accepted or else the obtained sign is rejected.

Input: message m, private and public keys (Q, E, D, N)

* CT1 = m\*Q mod N
* CT2 = m\*E mod N
* CT3 = m\*D mod N
* CT4 = m\*Q\*E mod N
* CT5 = m\*Q\*D mod N
* CT6 = m\*D\*E mod N (CT – Cipher-text)

Decryption: Decryption here gives the message. So if it matches then the sign obtained is accepted or else the obtained sign is rejected.

Input: CT1-CT6, Q, E, D, N

* DT1 = CT1\*E\*D mod N
* DT2 = CT2\*Q\*D mod N
* DT3 = CT3\*E\*Q mod N
* DT4 = CT4\*D mod N
* DT5 = CT5\*E mod N
* DT6 = CT6E\*Q mod N (DT- Decipher-text)

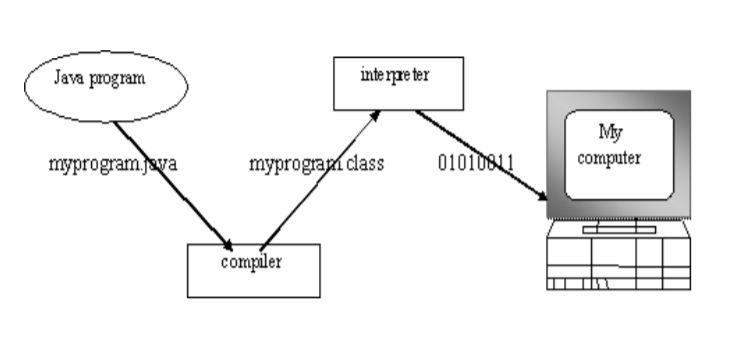
**CHAPTER 4**

**SOFTWARE DESCRIPTION**

4.1 JAVA PROGRAMMING LANGUAGE:

It is a Platform Independent. Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling’s office), was intended to replace C++, although the feature set better resembles that of Objective C. You can compile your Java program into byte codes on any platform that has a Java compiler

Function:



CHARATERISTICS:

Java is a high-level programming language that is all of the following:

* Simple & Object-oriented Distributed
* Interpreted & Robust
* Secure & Architecture-neutral
* Portable & High-performance
* Multithreaded & Dynamic

Java Virtual Machine:

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (JVM). Every Java interpreter, whether it’s a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

Fig 4.1

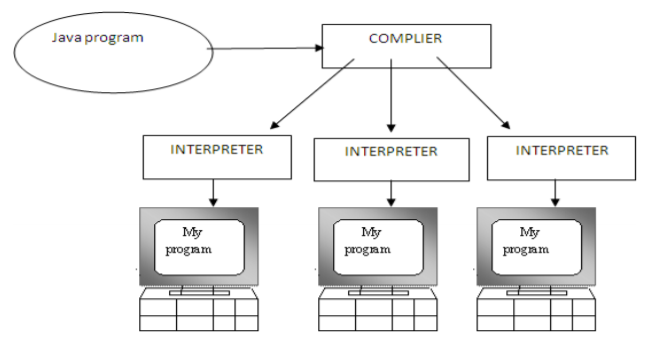


Fig 4.1

41

4.2 SYSTEM SPECIFICATIONS:

SOFTWARE REQUIREMENTS:

1. OPERATING SYSTEM: Windows XP or Higher
2. Languages used: Java (JSP, Servlet)
3. HTML Tools: JDK 1.7, Net Beans 7.0.1
4. SQL log Backend: My SQL

HARDWARE REQUIREMENTS:

1. PROCESSOR: Intel Pentium Dual Core 2.3 GHz
2. MOTHERBOARD: Intel 915GVSR chipset board
3. RAM: 1 GB (min)
4. HARD DISK DRIVE: 250 GB HDD.

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

**CHAPTER 5**

**IMPLEMENTATION**

**CODE :**

package com.mycompany.hybridcryptobs;

import java.io.\*;

import java.math.BigInteger;

import java.util.Random;

public class HybridcryptoBS

{

public static void main(String[] args)throws IOException

{

//\*\*\*\*KGS Start

long startTime1=System.currentTimeMillis();

long total1 = 0;

for (int i1 = 0; i1 < 10000000; i1++) total1 += i1;

BigInteger N,piN,d;

BigInteger one=new BigInteger("1");

BigInteger pi,qi,ei;

int bitLength = 1024; // INPUT bit length of primes P and Q;

Random rnd = new Random(); // create a random object

pi = BigInteger.probablePrime(bitLength, rnd);

qi = BigInteger.probablePrime(bitLength, rnd);

ei = BigInteger.probablePrime(bitLength, rnd); // Assigns probable Prime to p,q,e using bitLength and rnd

BigInteger M=new BigInteger("16"); //Original message

// BigInteger a=new BigInteger("21");

//BigInteger b=new BigInteger("12");

BigInteger M1=pi.multiply(qi).subtract(pi); //M=A\*B-P

BigInteger E=M1.add(pi);//E=M+A

BigInteger D=M1.add(qi);//E=M+B

BigInteger N1=((E.multiply(D)).subtract(pi)).divide(M1);//N=ED-P/M

BigInteger Q=pi.modInverse(N1);

//System.out.println("Q: " +Q);

long stopTime1 = System.currentTimeMillis();

long elapsedTime1 = stopTime1 - startTime1;

System.out.println("KGS time:");

System.out.println(elapsedTime1);

//Encryption

long startTime2 = System.currentTimeMillis();

long total2 = 0;

for (int i2 = 0; i2 < 10000000; i2++) total1 += i2;

//Input Message

BigInteger m1=new BigInteger("3");

BigInteger m2=new BigInteger("5");

//Encryption

BigInteger CT1=(m1.multiply(D)).mod(N1);//CT=M\*D MOD N

BigInteger CT2=(m2.multiply(D)).mod(N1);

//System.out.println("CT1: " +CT1);

//System.out.println("CT2: " +CT2);

long stopTime2 = System.currentTimeMillis();

long elapsedTime2 = stopTime2 - startTime2;

System.out.println("Encryption time:");

System.out.println(elapsedTime2);

//Encryption end

//Decryption

long startTime3 = System.currentTimeMillis(); long total3 = 0;

for (int i3 = 0; i3 < 10000000; i3++) total1 += i3;

//Decryption

BigInteger DT1=(Q.multiply(E).multiply(CT1)).mod(N1); //DT = CT\*E\*Q MOD N

BigInteger DT2=(Q.multiply(E).multiply(CT2)).mod(N1);

System.out.println("DT1:"+DT1);

System.out.println("DT2:"+DT2);

N=pi.multiply(qi); //N=P\*Q

piN=(pi.subtract(one)).multiply(qi.subtract(one));//PI(N) =(P-1)\*(Q-1)

d=ei.modInverse(piN);//D=E INVERSE PI(N)

//System.out.println("Private Key:"+d);

BigInteger CT;

//CT=M.modPow(e,N);

//System.out.println("Message:"+M);

//System.out.println("Cipher Text:"+CT);

//BigInteger DT;

//DT=CT.modPow(d,N);

//System.out.println("Decipher Text:"+DT);

BigInteger r=new BigInteger("5");

//Blind Factor r^e MOD N

BigInteger X;

X=(r.modPow(ei,N)).mod(N); //

//Blinded Message

BigInteger XM;

XM=(X.multiply(M)).mod(N); //BM = (BF\*M) MOD N

//System.out.println("Blinded Message XM:"+XM);

long stopTime3 = System.currentTimeMillis();

long elapsedTime3 = stopTime3 - startTime3;

System.out.println("Blinded Message time:");

System.out.println(elapsedTime3);

//Sign Generation

long startTime4 = System.currentTimeMillis();

long total4 = 0;

for (int i4 = 0; i4 < 10000000; i4++) total1 += i4;

BigInteger s; s=XM.modPow(d,N); //S = BM^d MOD N

//System.out.println("Signed value s:"+s);

long stopTime4 = System.currentTimeMillis();

long elapsedTime4 = stopTime4 - startTime4;

System.out.println("Sing Generation time:");

System.out.println(elapsedTime4);

long startTime5 = System.currentTimeMillis();

long total5 = 0;

for (int i5 = 0; i5 < 10000000; i5++) total1 += i5;

//Sign verification

BigInteger t1,t;

t1=(r.modInverse(N));

//System.out.println("r inverse: "+t1);

t=((r.modInverse(N)).multiply(s)).modPow(ei,N);//M'=S\*R^-MOD N

System.out.println("True sign t:"+t);

long stopTime5 = System.currentTimeMillis();

long elapsedTime5 = stopTime5 - startTime5;

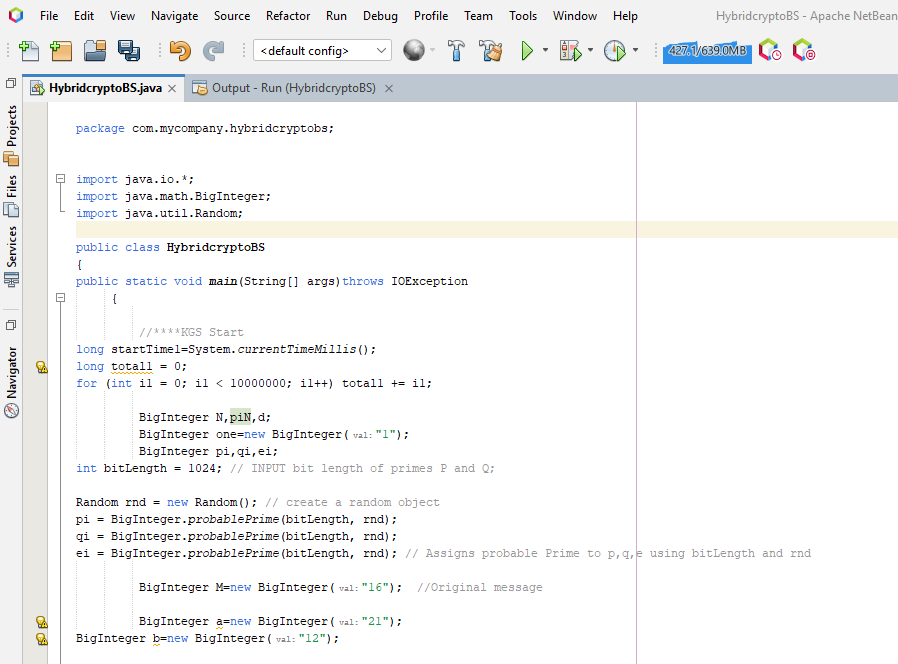
System.out.println("Blind verification time:");

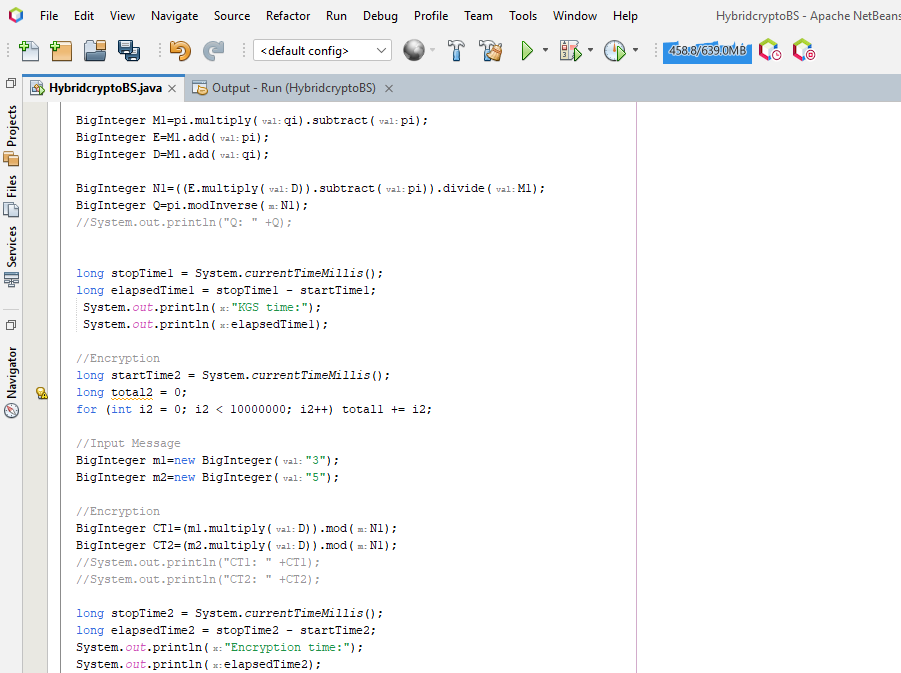
System.out.println(elapsedTime5);

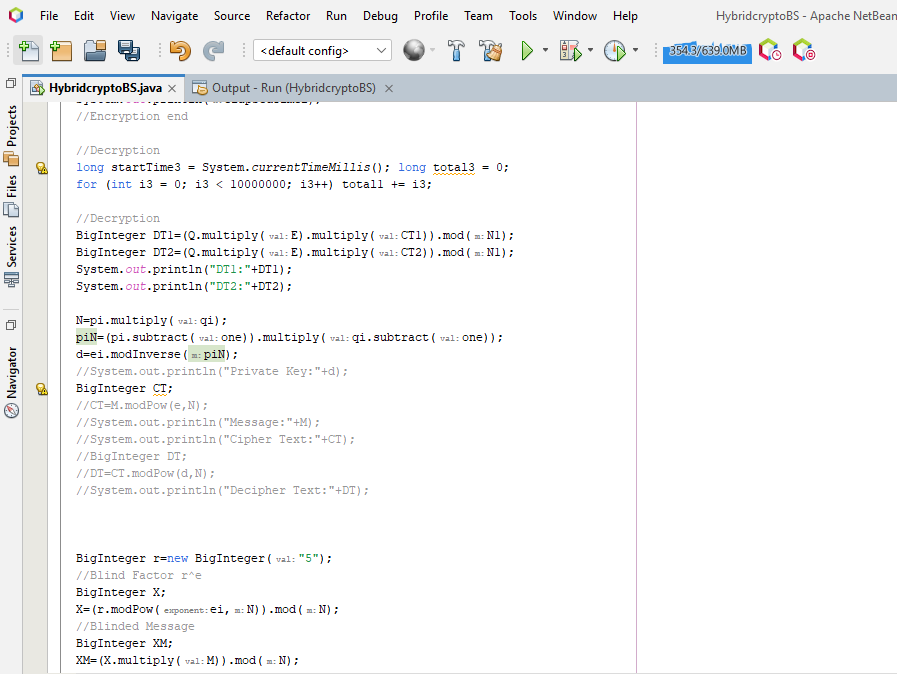
}

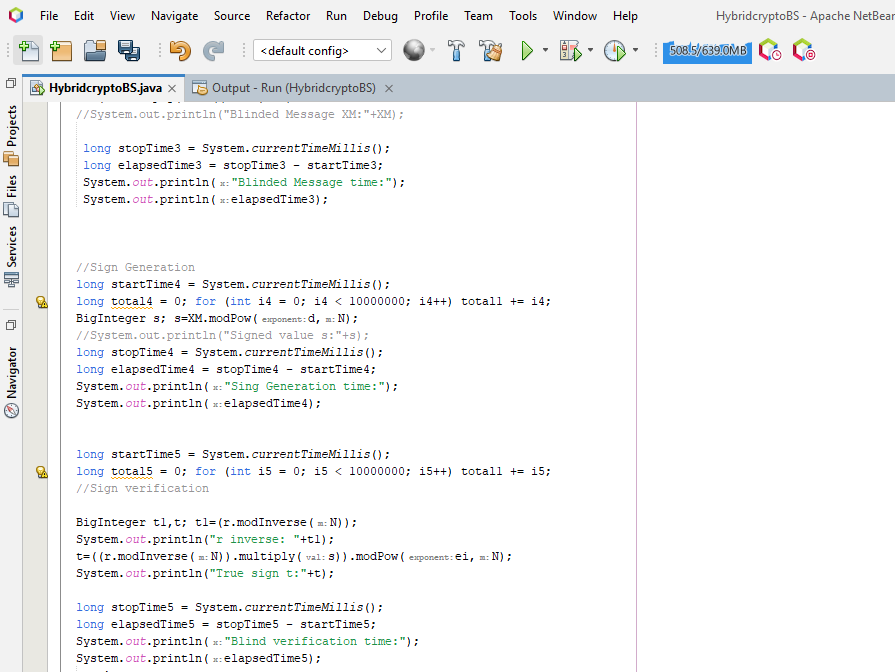
}

**SCREENSHOT :**

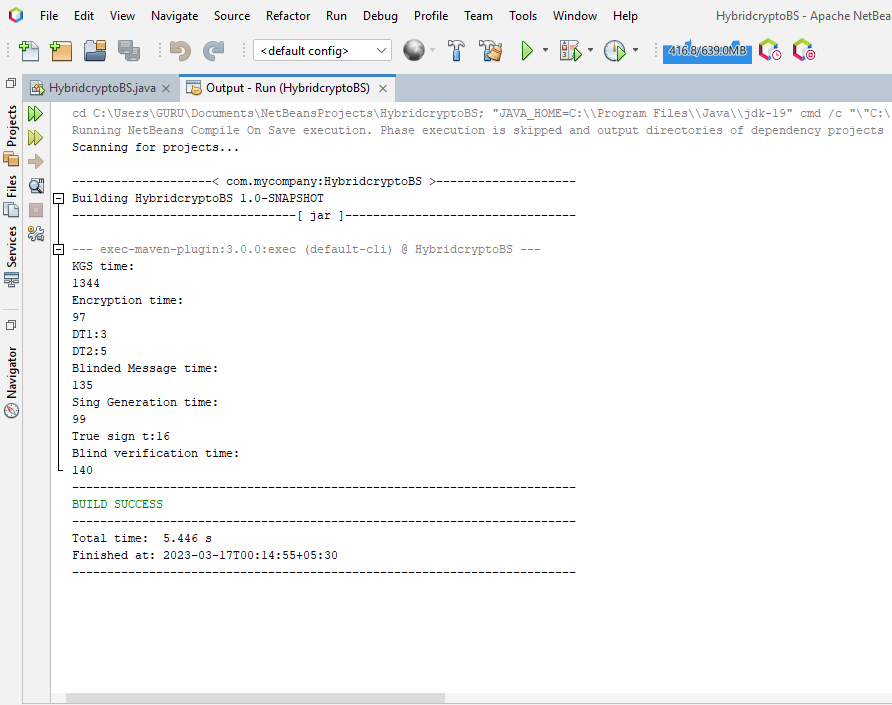
****



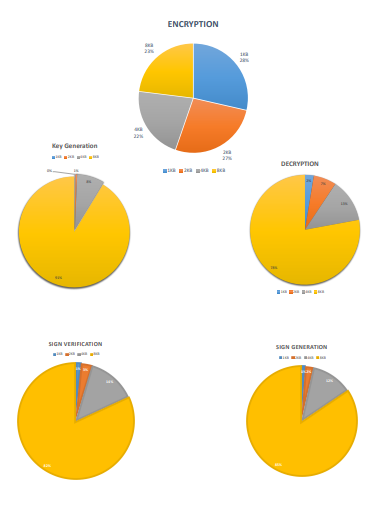


****

**OUTPUT :**



**TIME CONSUMPTION :**



**Chapter 7**

**Conclusion & Future Work:**

Blind Signature is a form of digital signature in which the content of a message is disguised (blinded) before it is signed. . Blind signatures are typically employed in privacy-related protocols where the signer and message author are different parties. Blind signatures can be used to authenticate the source of messages. When ownership of a blind signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. If a message is digitally signed, any change in the message will invalidate the signature. Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions. Future works are based on Chan-PKC’s decryption and encryption which gives exponential terms in Blind factor which can be more complex so that it will enhance the confidentiality and authentication and here the order of complexity is one so that it can work on any systems and in future it can be extended as a dual authentication process.

**References:**

1. S.A. Brands. Untraceable Off-line Electronic Cash Based on Secret-key Certificates. Latin 95.

2. M. Bellare and S. Micali. “How to Sign Given Any Trapdoor Function”. STOC 88.

3. M. Bellare and P. Rogaway. “Random Oracles are Practical: a Paradigm for Designing Efficient Protocols”. In Computer and Communication Security Conference, 1993.

4. M. Bellare and S. Goldwasser. “New Paradigms for Digital Signatures and Message Authentication Based on Non-Interactive Zero Knowledge Proofs”. Crypto 89 proceedings, pp. 194 -211

5. M. Bellare and P. Rogaway. “The Exact Security of Digital Signatures – How to Sign with RSA and Rabin”. Eurocrypt-96.

6. R. Canetti “De-mystifying Random Oracles” CRYPTO-97 (this proceedings).

7. R. Canetti, Y. Lindell, R. Ostrovsky, A. Sahai: “Universally composable two-party and multi-party secure computation”. STOC 2002: 494-503

8. D. Chaum. “Blind Signatures for Untraceable Payments”. Crypto-82.

9. D. Chaum, A. Fiat, and M. Naor. “Untraceable Electronic Cash”, Crypto-89.

10. C. Dwork and M. Naor. “An Efficient Existentially Unforgeable Signature Scheme and its Applications”. Crypto 94.

11. W. Diffie and M. Hellman. “New Directions in Cryptography”. IEEE Trans. on Inf. Theory, IT-22, pp. 644–654, 1976.

12. A. Fiat and A. Shamir. “How to Prove Yourself: Practical Solutions of Identification and Signature Problems, CRYPTO 86.

13. O. Goldreich.“Two Remarks Concerning the GMR Signature Scheme” MIT Tech. Report 715, 1986. CRYPTO 86.

14. O. Goldreich, S. Goldwasser, and S. Micali. “How to Construct Random Functions”. JASM V. 33 No 4. (October 1986) pp. 792-807.

57

15. S. Goldwasser, S. Micali and C. Rackoff, “The Knowledge Complexity of Interactive Proof-Systems”. SIAM J. Comput. 18 (1989), pp. 186-208; (also in STOC 85, pp. 291-304.)

16. O. Goldreich, S. Micali, and A. Wigderson. “How to Play Any Mental Game”. Proc. of 19th STOC, pp. 218-229, 1987.

17. S. Goldwasser, S. Micali, and R. Rivest. “A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks”. SIAM Journal of Computing Vol. 17, No 2, (April 1988), pp. 281-308.

18. Goldwasser S., and R. Ostrovsky “Invariant Signatures and Non-Interactive ZeroKnowledge Proofs are Equivalent” CRYPTO 92.

19. L.C. Guillou and J.J. Quisquater. “A Practical Zero-Knowledge Protocol Fitter to Security Microprocessor Minimizing Both Transmission and Memory”. EUROCRYPT 88.

20. M. Naor. “Bit Commitment Using Pseudo-Randomness”. Crypto-89.

21. I. Damgard, P. Landrock, and C. Pomerance, C. “Average Case Error Estimates for the Strong Probable Prime Test,” Mathematics of Computation, v. 61, No, 203, pp. 177-194, 1993. 22. A.J Menezes, P.C. Oorschot, and S.A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

23. H.C. Williams. “A p+1 Method of factoring”. Math. Comp. 39, 225-234, 1982.

24. D.E. Knuth, The Art of Computer Programming, Vol. 2, 3rd Ed., Addison-Wesley, 1998, Algorithm P, page 395.

25. R. Baillie and S.S. Wagstaff Jr., Mathematics of Computation, V. 35 (1980), pages 1391 – 1417.