LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5004NI Security in Computing

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2023 -24 Autumn

Student Name: Satyandra Kayastha

London Met ID: 22085599

College ID: NP01NT4S230016

Assignment Due Date: Monday 15th January 2024

Assignment Submission Date: Sunday 14th January 2024

Word Count (Where Required): 4304

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

This report details the study, creation, and testing of the new RSA cryptographic system as part of the "Security in Computing" curriculum. The major goal was to address data security challenges by constructing a strong solution, and tasks included in-depth cryptography investigation, the selection of RSA as the basic method, and the introduction of novel mathematical operations. Testing across several situations revealed insights into performance and indicated potential improvements, while the review highlighted RSA's strengths and weaknesses. The official presentation highlighted major aspects and contributions. The experience not only increased our understanding of cryptographic systems, but it also placed us in a position to contribute to continuing conversations about data security and cryptographic advancements.

Table of Contents

Table of figure

1. Introduction

In the field of information systems, security involves protecting against unauthorized entry, utilization, disclosure, disruption, modification, or elimination of both information and systems. It encompasses a wide array of techniques and approaches aimed at safeguarding data, applications, and infrastructure from cyber threats and weaknesses. (CISCO, 2022)

The three key aspects of information security are:

- Confidentiality
- Integrity
- Availability

Figure 1: CIA Triad

a. Confidentiality

Confidentiality guarantees that sensitive information can only be accessed by authorized individuals and systems. This includes implementing security measures such as access control, encryption, and data masking. Think of it like a locked file cabinet where only authorized personnel possess the key. (nist, www.nist.com, 2022)

b. Integrity

The concept of integrity guarantees that information remains accurate and intact, free from any unauthorized changes or damage. This is achieved through different methods like data validation, checksums, and digital signatures. Visualize a document with a seal that clearly indicates whether it has been tampered with, providing assurance that its contents remain unchanged. (nist, nist, 2022)

c. Availability

Authorized users can have confidence in the fact that information and systems will always be accessible to them when needed. This includes the use of backup systems, plans for recovering from disasters, and efforts to improve performance. Imagine a well-maintained bridge that is always ready for authorized vehicles to cross. (ashushrm, 2023)

1.1 Cryptography

Cryptography involves the act of hiding or encoding data, ensuring that only the intended recipient possessing the decryption key can decode it. Its name originates from the Greek terms "kryptós," signifying hidden, and "graphein," signifying to write. In essence, cryptography is the method of securely transmitting information. (kaspersky, 2023)

the key terminologies of cryptography are:

- Encryption: The use of an algorithm and a key to transform plaintext to cipher text.
- Decryption: Using the appropriate key, the process of turning cipher text back into plaintext.
- Cipher text: The plain text had been encrypted.
- Plain text: information in its original, readable form.
- Key: A bit of info that is used to encrypt and decode data.
- Symmetric encryption: The same key is used for both encryption and decryption. Think it as a lock with a single key.

- Asymmetric encryption: It utilizes two separate keys for encryption and decryption: a public key for encryption and a private key for decryption. Think it as a mailbox with a public slot for anybody to drop letters into, but only the owner has the key to open it.

1.2 History of cryptography

Cryptography, which is derived from the Greek words for "hidden writing," is the study of hiding communicated information such that only those who were meant to receive it is able to understand it. Since ancient times, practically all significant civilizations have used hidden signals to communicate with one another. (Schneider, 2024)

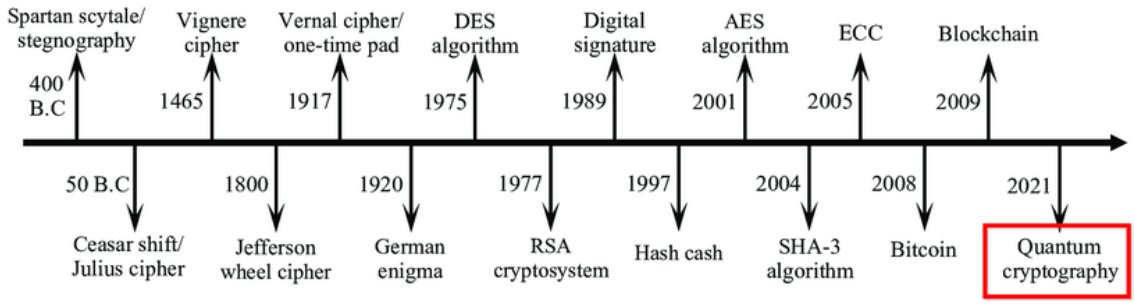

Figure 2: Timeline Of Cryptography

Ancient cryptography:

100-44 BC: Julius Caesar is credited with inventing the Caesar Cipher, a substitution cipher in which each letter of the plaintext is replaced by a different letter determined by moving a set number of letters forward or backward within the Latin alphabet, to share secure communications within the Roman army. The private key in this symmetric key cryptosystem is the exact steps and direction of the letter transposition.

Medieval cryptography:

1467 AD: Considered the father of modern cryptography, Leon Battista Alberti's work most clearly examined the use of polyphonic cryptosystems, or ciphers containing several alphabets, as the strongest type of encryption in the Middle Ages.

Modern Cryptography:

1913 AD: The start of World War I at the turn of the twentieth century witnessed a significant surge in both cryptology for military communications and cryptanalysis for codebreaking. The effectiveness of English cryptologists in breaking German telegraph codes resulted in critical victories for the Royal Navy.

1918 AD: Following the war, German cryptologist Arthur Scherbius created the Enigma Machine, an upgraded version of Hebern's rotor machine that employed rotor circuits to both encode and decode plaintext. The Enigma Machine, which was widely used by the Germans during and during WWII, was regarded suited for the highest degree of top-secret encryption. Decoding a message encrypted with the Enigma Machine, like Hebern's Rotor Machine, needed sophisticated sharing of machine calibration settings and private keys, which were vulnerable to espionage and finally led to the Enigma's demise.

1939-45 AD: When World War II broke out, Polish codebreakers escaped Poland and joined several important and famous British mathematicians, including the father of modern computers, Alan Turing, to crack the German Enigma cryptosystem, a key breakthrough for the Allies. Turing's work, in particular, laid the groundwork for most of the basic theory underpinning algorithmic computations.

1975 AD: IBM researchers working on block ciphers created the Data Encryption Standard (DES), the first cryptosystem approved for use by the US government by the National Institute of Standards and Technology (formerly known as the National Bureau

of Standards). While the DES was powerful enough to defeat even the most powerful computers of the 1970s, its short key length makes it unsuitable for current uses, but its architecture was and continues to be crucial in the evolution of cryptography.

1976 AD: The Diffie-Hellman key exchange method was developed by researchers Whitfield Hellman and Martin Diffie for securely transferring cryptographic keys. This opened the way for a new type of encryption known as asymmetric key algorithms. Because they no longer rely on a shared private key, these algorithms, also known as public key cryptography, provide an even greater level of anonymity. In public key cryptosystems, each user has their own private secret key that, for increased security, works in tandem with a common public key.

1977 AD: Adi Shamir and Leonard Adleman present the RSA public key cryptosystem on Rivest, one of the earliest encryption systems for safe data transfer that is still in use today. RSA public keys are generated by multiplying big prime integers, which are prohibitively difficult to factor without previous knowledge of the private key used to generate the public key.

2001 AD: In response to advances in computer power, the DES encryption algorithm was superseded with the more resilient Advanced Encryption Standard (AES). The AES is a symmetric cryptosystem like the DES, but it employs a significantly longer encryption key that cannot be broken by contemporary technology.

1.3 Symmetric Encryption

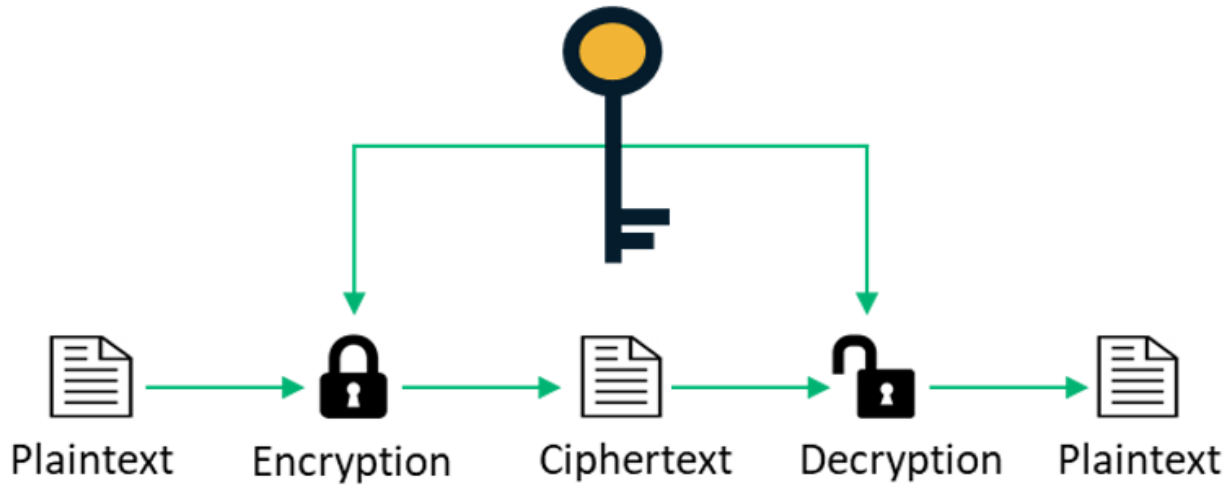

Figure 3: Symmetric Encryption

- The same key is used for both encryption and decryption in symmetric encryption.

Symmetric encryption is similar to a secret code shared by two friends. Assume you and a friend wish to send a private message that no one else can read. You come to an agreement on a secret codebook in which each letter in your message is substituted with another letter or symbol. The message appears to your buddy as gibberish, but with the codebook, they can quickly translate it back to the original message. (Mukherjee, 2020)

Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) are examples of common symmetric encryption methods.

1.4 Asymmetric Encryption

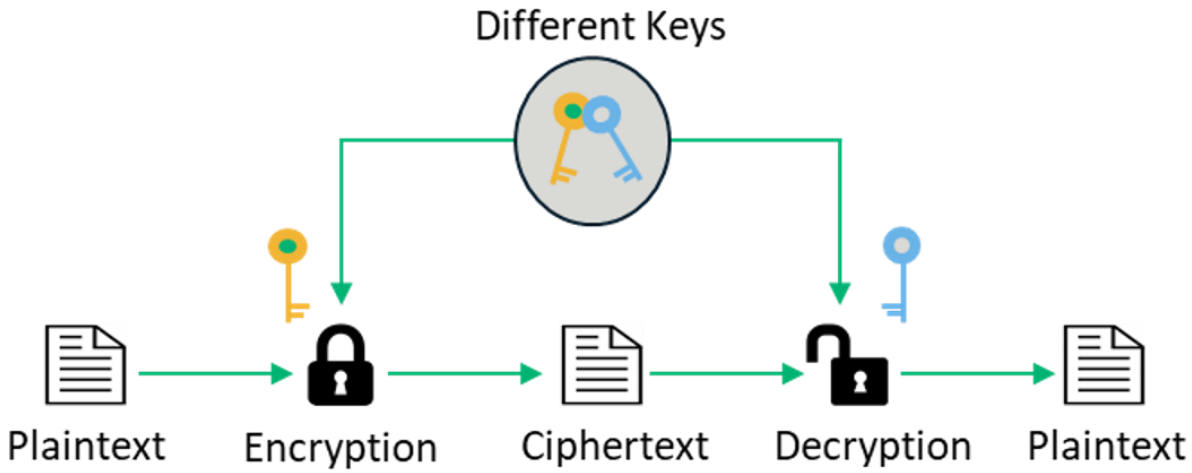

Figure 4: Asymmetric Encryption

- Asymmetric encryption, also referred to as public-key encryption, utilizes a combination of two keys: public key and private key.

The public key is open to the public and is used for encryption, but the private key is kept private and is used for decryption. Asymmetric encryption is similar to having a special mailbox with two keys. To place messages in the mailbox, you offer everyone a spare key (the public key), but only you have the master key (the private key). Anyone can send you a message, but only you have the ability to read it. Even if someone discovers the spare key, they won't be able to unlock the mailbox without your master key. (Okta, 2023)

RSA (Rivest-Shamir-Adleman), Diffie-Hellman key exchange, and Elliptic Curve Cryptography (ECC) are examples of popular asymmetric encryption method.

1.5 Aim

The coursework's aim is to provide students with the ability to study, design, and test an innovative cryptographic system, creating a greater understanding of cryptography principles while encouraging creativity in coming up with solutions to improve security of information and confidentiality.

1.6 Objective

- To understand cryptography by researching history, symmetric and asymmetric algorithm.
- To develop a new cryptographic algorithm which include mathematical and logical operation.
- To test the new cryptography algorithm for improvement.
- To understand the strengths and weakness of new cryptography algorithm.

2. Background

2.1 RSA algorithm

The RSA algorithm, which was created by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978, is a type of asymmetric cryptography algorithm. This means that it relies on two distinct keys - a public key and a private key - that are mathematically connected. As the names imply, the public key is meant to be shared openly, while the private key must be kept confidential and not disclosed to anyone. (teachcomputerscience, 2023)

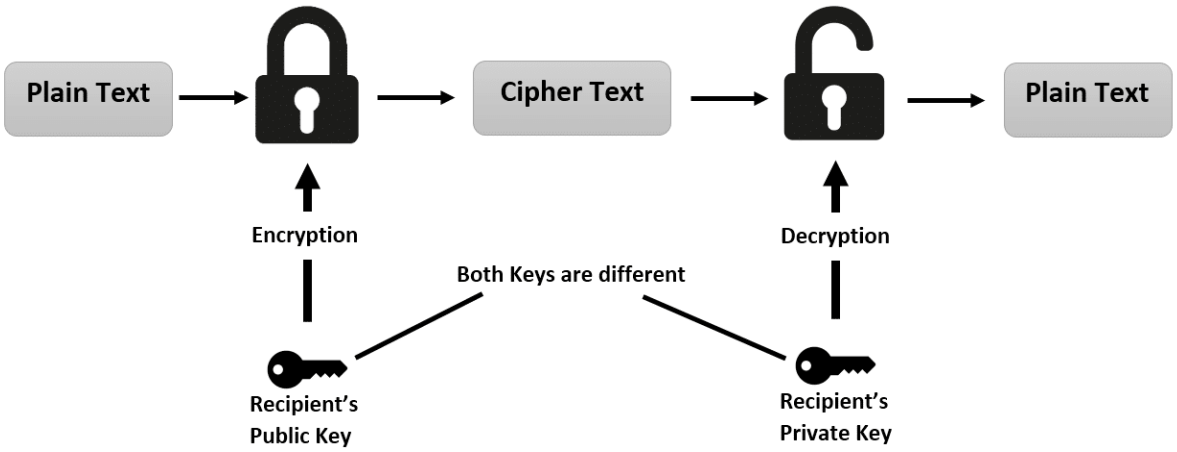

Figure 5: RSA

(AlHussein, 2021)

Key generation

- step 1: take two large prime numbers "**p**" and "**q**" which are appointed, where p!=q.
- step 2: calculate the value, **n = p × q.**
- step 3: calculate the value with, **Ø(n) = (p – 1) × (q – 1).** where Ø is Euler's totient function.
- step 4: select a public key exponent "**e**" such that, **1 < e < n** and **gcd(e, Ø(n)) = 1**.
- Step 5: compute the secret private key exponent "**d**" such that**, 1 < d < Ø(n)** and **e × d mod Ø(n) = 1.**
- Step 6: public key = (n, e)
- Step 7: private key = (n, d)

Encryption

- Step 1: Take the public key – (n, e).
- Step 2: denote plain text message as positive integer m.
- Step 3: calculate cipher text, c = m^e mod n.

Decryption

- Step 1: take the private key – (n, d).
- Step 2: calculate plain text, m = c^d mod n.

Example of RSA algorithm:

Step 1: asymmetric key generation,

a. Let's, take the values of p and q are 5 and 7
   p = 5, q = 7

b. n = p × q = 35
c. Ø(n) = (p – 1) x (q – 1) = 24
d. e = 5, since gcd (5, 24) = 1 && 1 < 5 < 35
e. d = 5, since e x d mod Ø(n) = 1 …. [i.e. 5 x 5 mod 24 = 1]

step 2: RSA encryption,

a. c = m^e mod n
b. c = 3^5 mod 35 = 33 (cipher text)

step 3: RSA decryption,

a. m = c^d mod n
b. m = 33^5 mod 35 = 3 (plain text)

Advantages of RSA algorithm

- RSA relies on the mathematical complexity of factoring large prime numbers, which poses a significant challenge for potential attackers.
- RSA uses two keys, one public and one private. This enables safe data communication without the need for a shared secret key.
- RSA is a very simple implementation when compared to more complicated cryptographic algorithms, making it easy to integrate into multiple systems.
- RSA can be used to generate digital signatures that validate a message's validity and integrity.

Disadvantages of RSA algorithm

- RSA operations are computationally complex, particularly with bigger key sizes. When compared to other encryption techniques, this might result in worse performance.
- The security of RSA is strongly dependent on key size. Larger key sizes provide more security, but they also need more storage space and processing capacity.
- RSA is inefficient for encrypting large volumes of data. For mass encryption, it is frequently used in combination with symmetric-key techniques.
- Managing public and private keys safely may be difficult, especially in big businesses with numerous users.

3. Development

3.1 Modified RSA Algorithm

The term "modified cryptography" often refers to the creation or use of cryptographic algorithms or systems that are not part of commonly acknowledged standards or protocols. It involves designing cryptographic solutions that are adapted to individual requirements or scenarios. (dhakar, 2012)

The new modified RSA algorithm is named as **Z-RSA.**

3.2 Modified RSA Key generation:

a. Selecting Primes (p, q, r, s, t):

   In the initial phase of my key generation, I differ from the typical RSA technique by picking five prime numbers rather than the traditional two (p and q). RSA security often depends on the difficulty of factoring the product of two big primes.

b. Computing modulus (n):

   The modulus (n) is calculated by multiplying the product of the primes (p, q, r, s, t). the modulus in traditional RSA is created by multiplying only two primes. Increasing this to five primes increases the computational difficulty of the modulus calculation.

c. Computing Euler's totient function Ø(n):

   The Euler's totient function Ø(n) is determined as the prime factors selected minus one. This phase is consistent with traditional RSA key generation. As Ø(n) is required for calculating the public and private exponents.

d. Selecting public exponent (e):

The procedure for determining the public exponent (e) differs significantly from the traditional RSA technique. The condition is $p > e > Ø(n)$ AND $p < e < Ø(n)$ present a new requirement for locating an e that is coprime to both “n” and “Ø(n)”. I have use the “AND” logical operation in my RSA.

e. Calculating $z = (e+5) \times 2$ and computing public exponent:

The calculation of z as $(e+5) \times 2$ maybe unusual addition to the RSA key generation but the purpose of this procedure for implementing mathematical operation as per required changes in given coursework task 3 of security in computing. The objective of this process for as modified public key.

f. Computing private exponent (d):

The calculation of the private exponent (d) follows the traditional RSA approach, making sure that, $1 < d < Ø(n)$ and $e*d \bmod Ø(n) = 1$. This phase is crucial for establishing the association between the public keys and private keys and sustaining the RSA algorithm security features.

g. Public Key and Private Key Representation:

The public key can be represented as (n, s) differs from the traditional norm in which the public key typically express as (n, e). the private key remains as (n, d) which is from traditional RSA. Hence, the addition of s in the public key provides the unusual aspect to key pair representation.

Modified RSA Key generation algorithm/ steps:

1. Select prime number p, q, r, s, t
2. Compute n = p * q * r * s * t
3. Compute Ø(n) = (p – 1) x (q – 1) x (r – 1) x (s – 1) x (t – 1)
4. Collect “e” with the condition p > e > Ø(n) AND p < e < Ø(n), coprime to n and Ø(n).
5. Calculate z = (e + 5) * 2
6. Compute the secret private key exponent “d” such that, 1<d< Ø(n) and
   e*d mod Ø(n) = 1.
7. Public key = (n, z)
8. Private key = (n, d)

3.3 Modified RSA Encryption Algorithm:

- **Step 01. Acquire the recipient’s Public Key (n, z).**
- **Step 02. Denote Plain Text message as a positive integer M.**
- **Step 03. Compute Cipher Text as, C = M^(s/2) – 5 mod n.**

The RSA cryptosystem's security is based on two mathematical problems. the issue of massive factoring numbers. The problem of attempting all potential private keys is known as a mathematical attack, while the challenge of trying all possible private keys is known as a brute force assault. To boost security, this approach introduces the Modified RSA Encryption technique (MREA), an innovative cryptographic technique based on additive homomorphic characteristics. https://ieeexplore.ieee.org/document/6168406

The formula I have include for computing the cipher text is much different from the traditional RSA encryption formula, which was c= m^e mod n for public key (n,e). my formula use **C = M^(s/2) – 5 mod n.**

3.4 Modified RSA Decryption Algorithm:

- **Step 01. Acquire the recipient's Private Key – (n, d).**
- **Step 02. Compute Plain Text as, M = C^d mod n.**

For the decryption procedure, I implemented the traditional RSA decryption technique. It involves required private key, which comprises the modulus "n" and the private exponent "d". In order to decrypt a cipher text "C" and get original plaintext "M", the operation we get, **M = C^d mod n.**

4. Flowchart

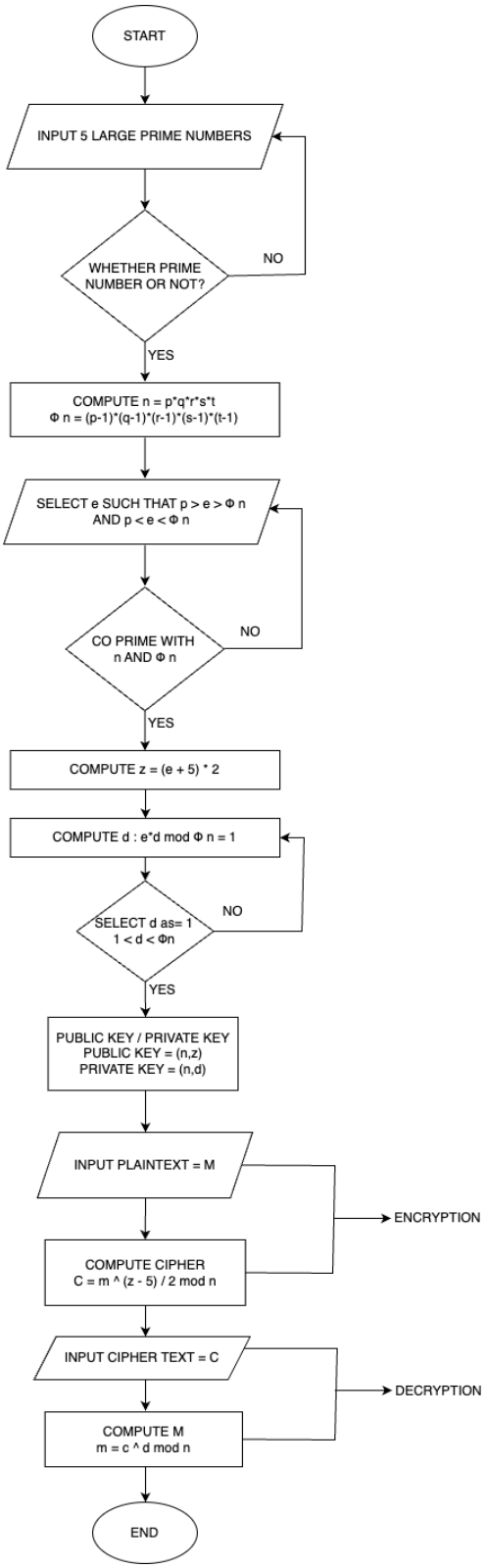

Figure 6: Flowchart Of Z-RSA

5. Testing

5.1 Test 1:

a. Key generation:

1. Select prime numbers:

   p = 17, q = 19, r = 23, s = 29, t = 31

2. Compute n:

   n = p * q * r * s * t = 17 * 19 * 23 * 29 * 31 = 349691

3. Compute Euler's totient function Ø(n):

Ø(n) = (p - 1) * (q - 1) * (r - 1) * (s - 1) * (t - 1)

Ø(n) = 16 * 18 * 22 * 28 * 30 = 583200

4. Select public key exponent e:

Let's choose e = 499 (a random coprime to both n and Ø(n))

5. Calculate z:

z = (e + 5) * 2 = (499 + 5) * 2 = 1008

6. Compute private key exponent d:

We need to find d such that e * d mod Ø(n) = 1.

Let's choose d = 295679 (a random, but satisfies the condition)

7. Public Key:

(n, z) = (349691, 1008)

8. Private Key:

(n, d) = (349691, 295679)

b. Encryption

Suppose, M = 123

- C = M^(z/2) – 5 mod n
- C = 123^(1008/2) – 5 mod 349691
- C = 154536

c. Decryption

Here, C = 341272

- M = C^d mod n
- M = 154536^295679 mod 349691
- M = 123

5.2 Test 2:

a. Key generation:

1. Select prime numbers:
   p = 41, q = 43, r = 47, s = 53, t = 59

2. Compute n:
   n = p * q * r * s * t = 41 * 43 * 47 * 53 * 59 = 369216059

3. Compute Euler's totient function Ø(n):
   Ø(n) = (p - 1) * (q - 1) * (r - 1) * (s - 1) * (t - 1)
   Ø(n) = 40 * 42 * 46 * 52 * 58 = 25082240

4. Select public key exponent e:
   Let's choose e = 1229 (a random coprime to both n and Ø(n))

5. Calculate z:
   z = (e + 5) * 2 = (1229+ 5) * 2 = 2468

6. Compute private key exponent d:
   We need to find d such that e * d mod Ø(n) = 1.
   Let's choose d = 55153 (random, but satisfies the condition)

7. Public Key:
   (n, z) = (369216059, 2468)

8. Private Key:
   (n, d) = (369216059, 55153)

b. Encryption

Suppose, M = 456

- C = M^(z/2) – 5 mod n
- C = (456^1234 - 5) mod 369216059
- C = 307312637

c. Decryption

Here, C = 307312637

- M = C^d mod n
- M = 307312637^55153 mod 369216059
- M = 456

5.3 Test 3:

a. Key generation:

1. Select prime numbers:
   p = 61, q = 67, r = 71, s = 73, t = 79

2. Compute n:
   n = p * q * r * s * t = 61 * 67 * 71 * 73 * 79 = 174366969

3. Compute Euler's totient function Ø(n):
   Ø(n) = (p - 1) * (q - 1) * (r - 1) * (s - 1) * (t - 1)
   Ø(n) = 60 * 66 * 70 * 72 * 78 = 14515200

4. Select public key exponent e:
   Let's choose e = 1073 (a random coprime to both n and Ø(n))

5. Calculate z:
   z = (e + 5) * 2 = (1073 + 5) * 2 = 2166

6. Compute private key exponent d:
   We need to find d such that e * d mod Ø(n) = 1.
   Let's choose d = 677729 (random, but satisfies the condition)

7. Public Key:
   (n, z) = (174366969, 2166)

8. Private Key:
   (n, d) = (174366969, 677729)

b. Encryption

Suppose, M = 789

- C = M^(z/2) – 5 mod n
- C = 789^(2166/2) - 5 mod 174366969
- C = 114285796

c. Decryption

Here, C = 114285796

- M = C^d mod n
- M = 114285796^677729 mod 174366969
- M = 789

5.4 Test 4:

a. Key generation:

1. Select prime numbers:
   p = 43, q = 47, r = 53, s = 59, t = 61

2. Compute n:
   n = p * q * r * s * t = 43 * 47 * 53 * 59 * 61 = 59304719

3. Compute Euler's totient function Ø(n):
   Ø(n) = (p - 1) * (q - 1) * (r - 1) * (s - 1) * (t - 1)
   Ø(n) = 42 * 46 * 52 * 58 * 60 = 60040320

4. Select public key exponent e:
   Let's choose e = 733 (a random coprime to both n and Ø(n))

5. Calculate z:
   z = (e + 5) * 2 = (733 + 5) * 2 = 1476

6. Compute private key exponent d:
   We need to find d such that e * d mod Ø(n) = 1.
   Let's choose d = 415397 (random, but satisfies the condition)

7. Public Key:
   (n, z) = (59304719, 1476)

8. Private Key:
   (n, d) = (59304719, 415397)

b. Encryption

Suppose, M = 1011

- C = M^(z/2) – 5 mod n
- C = 1011^738 - 5) mod 59304719
- C = 29388287

c. Decryption

Here, C = 29388287

- M = C^d mod n
- M = (29388287^415397) mod 59304719
- M = 1011

5.5 Test 5:

a. Key generation:

1. Select prime numbers:
   p = 71, q = 73, r = 79, s = 83, t = 89

2. Compute n:
   n = p * q * r * s * t = 71 * 73 * 79 * 83 * 89 = 216781679

3. Compute Euler's totient function Ø(n):
   Ø(n) = (p - 1) * (q - 1) * (r - 1) * (s - 1) * (t - 1)
   Ø(n) = 70 * 72 * 78 * 82 * 88 = 263781120

4. Select public key exponent e:
   Let's choose e = 1729 (a random coprime to both n and Ø(n))

5. Calculate z:
   z = (e + 5) * 2 = (1729 + 5) * 2 = 3468

6. Compute private key exponent d:
   We need to find d such that e * d mod Ø(n) = 1.
   Let's choose d = 1690981 (random, but satisfies the condition)

7. Public Key:
   (n, z) = (216781679, 3468)

8. Private Key:
   (n, d) = (216781679, 1690981)

b. Encryption

Suppose, M = 1234

- C = M^(z/2) – 5 mod n
- C = (1234^1734 - 5) mod 216781679
- C = 44690036

c. Decryption

Here, C = 44690036

- M = C^d mod n
- M = (44690036^1690981) mod 216781679
- M = 1234

6. Evaluation

6.1 Strength of Z-RSA:

a. Increased prime numbers:

The use of five prime numbers in key generation could potentially improve security by making factoring the modulus more difficult. This adjustment may provide toughness to traditional prime factorization-based attacks.

b. Unique public exponent selection:

An unusual component is introduced by the requirement for selecting the public exponent. If the specified condition improves resistance to certain threats, it may lead to increased security.

c. Innovative cipher text calculation:

The cipher text is calculated using a unique formula: C = M(s/2) - 5 mod n. This modify might be considered a strength if it adds new security measures or defends against particular sorts of attacks.

d. **Efficient for small data:**

Asymmetric encryption techniques, such as Z-RSA, are computationally difficult for large amounts of data yet efficient for small amounts of data. This efficiency is especially beneficial in situations when data capacity is restricted.

e. **No need for shared secrets:**

Unlike symmetric-key cryptography, which requires communication parties to disclose a common secret, Z-RSA does not require a shared secret. This characteristic makes key management in distributed systems easier.

6.2 Weakness of Z-RSA

a. **Randomness in key generation**

The quality of the random prime numbers picked during key creation determines the strength of RSA. If the randomness is corrupted or predictable, the algorithm's security is affected.

b. **Performance:**

For encrypting large volumes of data, Z-RSA is often slower than symmetric key techniques. To overcome this, hybrid techniques are frequently utilized, in which RSA is used for key exchange and a symmetric cipher is used for bulk data encryption.

c. **No perfect forward confidentiality:**

Z-RSA does not guarantee complete forward confidentiality. If an attacker acquires the private key, he or she will be able to decode all previous and future conversations encrypted with that key.

d. **Key Size:**

The security of Z-RSA relates to the size of the key. Longer key lengths are necessary to maintain the same level of security as computing power

increases Because of the modulus's multiple prime factors, this approach may require bigger key sizes.

e. **Implementation errors:**
Vulnerabilities can be introduced by bugs and technical errors in cryptography libraries or unique implementations. To avoid such problems, it is critical to use well-established and audited libraries.

6.3 Application Area for Z-RSA

My Z-RSA algorithm may be used in real-world settings like as secure communication, digital signatures, key exchange, and other cryptographic applications. It can help to build secure channels between parties, sign and verify digital messages, securely exchange keys, secure file transfers, and contribute to protocols like SSH and SSL/TLS. However, for crucial applications, it is necessary to take caution and choose well-established cryptographic libraries or protocols, as they are subjected to rigorous security inspection and testing by the cryptographic community. While instructional or research-oriented, custom algorithms may lack the same level of validation and may represent security problems in real implementations.

7. Conclusion

In conclusion, the development and assessment of the RSA cryptographic system for the "Security in Computing" coursework gave useful insights into the complexities of current encryption algorithms. The road to implement RSA has been both helpful and hard, involving in-depth research, algorithm selection, innovative development, thorough testing, and thorough evaluation.

Cryptography will continue to grow in importance as a security measure in our digitalized and technologically advanced society, where automated information resources are

increasing. Access control and data security will need to be strengthened in electronic networks used for banking, retail, inventory control, benefit and service delivery, information storage and retrieval, distributed processing, and government applications.

8. Bibliography

AlHussein, A. (2021). *www.github.com*. Retrieved from https://github.com/AbdullahAlhussein/RSA-Algorithm.

ashushrm. (2023). *geeksforgeeks*. Retrieved from www.geeksforgeeks.org: https://www.geeksforgeeks.org/availability-in-information-security/

CISCO. (2022). *www.cisco.com*. Retrieved from https://www.cisco.com: https://www.cisco.com/c/en/us/products/security/what-is-information-security-infosec.html

dhakar, r. s. (2012). *IEEE*. Retrieved from www. eeexplore.ieee.org: https://ieeexplore.ieee.org/document/6168406

kaspersky. (2023). *www.kaspersky.com*. Retrieved from https://www.kaspersky.com/resource-center/definitions/what-is-cryptography: https://www.kaspersky.com/resource-center/definitions/what-is-cryptography

Mukherjee, L. (2020). *www.sectigostore.com*. Retrieved from https://sectigostore.com/blog/5-differences-between-symmetric-vs-asymmetric-encryption/.

nist. (2022). *nist*. Retrieved from https://csrc.nist.gov: https://csrc.nist.gov/glossary/term/integrity#:~:text=Definitions%3A,information%20non%2Drepudiation%20and%20authenticity.

nist. (2022). *www.nist.com*. Retrieved from https://csrc.nist.gov: https://csrc.nist.gov/glossary/term/confidentiality

Okta. (2023). *www.okta.com*. Retrieved from www.okta.com: https://www.okta.com/identity-101/asymmetric-encryption/

Schneider, J. (2024). *www.ibm.com*. Retrieved from https://www.ibm.com/blog/cryptography-history/.

teachcomputerscience. (2023). *www.teachcomputerscience.com*. Retrieved from https://teachcomputerscience.com: https://teachcomputerscience.com/asymmetric-encryption/

9. Appendices

a. MREA (Modified RSA Encryption Algorithm)

Published in: 2012 Second International Conference on Advanced Computing & Communication Technologies

Presented by: Ravi Shankar Dhakar; Amit Kumar Gupta; Prashant Sharma

The paper wrote by them is given below:

2012 Second International Conference on Advanced Computing & Communication Technologies

Modified RSA Encryption Algorithm (MREA)

Ravi Shankar Dhakar
Sr. Lecturer, GIET,
Kota, Rajasthan, India
ravi.dhakar83@gmail.com

Amit Kumar Gupta
Asst. Prof. , SBCET,
Jaipur, Rajasthan, India
cseprof_amit@*rediffmail.com*

Prashant Sharma
Lecturer, MIT,
Kota, Rajasthan, India
Prashant_harmony@*rediffmail.com*

Abstract
In asymmetric key cryptography, also called Public Key cryptography, two different keys (which forms a key pair) are used. One key is used for encryption & only the other corresponding key must be used for decryption. No other key can decrypt the message, not even the original (i.e. the first) key used for encryption. The beauty of this scheme is that every communicating party needs just a key pair for communicating with any number of other communicating parties. Once some one obtains a key pair, he /she can communicate with any one else. RSA is a well known public-key cryptography algorithm. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. The security of the RSA cryptosystem is based on two mathematical problems: the problem of factoring large numbers know mathematical attack and the problem of trying all possible private keys know brute force attack. So to improve the security, this scheme presents a new cryptography algorithm based on additive homomorphic properties called *Modified RSA Encryption Algorithm* (MREA). MREA is secure as compared to RSA as it is based on the factoring problem as well as decisional composite residuosity assumptions which is the intractability hypothesis. The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m_1 and m_2, one can compute the encryption of $m_1 + m_2$. This scheme also presents comparison between RSA and MREA cryptosystems in terms of security and performance.

Index Terms: Encryption, Public key, Private key, Security, RSA, Homomorphic

I. INTRODUCTION

To solve the problem of secure key management of Symmetric key cryptography, Diffie and Hellman introduced a new approach to cryptography and, in effect, challenged cryptologists to come up with a cryptographic algorithm that met the requirements for public-key systems. Public key cryptography uses a pair of related keys, one for encryption and other for decryption. One key, which is called the private key, is kept secret and other one known as public key is disclosed and this eliminate the need for the sender and the receiver to share secret key. The only requirement is that public keys are associated with the users in a trusted (authenticated) [8]manner through a public key infrastructure (PKI). The public key cryptosystems are the most popular, due to both confidentiality and authentication facilities [1].The message is encrypted with public key and can only be decrypted by using the private key. So, the encrypted message cannot be decrypted by anyone who knows only the public key and thus secure communication is possible. In a public-key cryptosystem, the private key is always linked mathematically to the public key. Therefore, it is always possible to attack a public-key system by deriving the private key from the public key. The defense against this is to make the problem of deriving the private key from the public key as difficult as possible. Some public-key cryptosystems are designed such that deriving the private key from the public key requires the attacker to factor a large number. The *RSA* and *MREA* public key cryptosystems [3] are the best known examples of such a system. This paper presents a hybrid cryptography algorithm which is based on the additive homomorphic properties called a *Modified RSA Encryption Algorithm(MREA).*

A. Homomorphic Properties

A notable feature of the MREA cryptosystem is its homomorphic properties. As the encryption function is additively homomorphic, the following identities can be described:

i. Homomorphic addition of plaintexts

The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts,

$$D(E(m_1,r_1) \cdot E(m_2,r_2) \mod n^2) = m_1 + m_2 \mod n.$$

The product of a ciphertext with a plaintext raising g will decrypt to the sum of the corresponding plaintexts,

$$D(E(m_1,r_1) \cdot g^{m2} \mod n^2) = m_1 + m_2 \mod n.$$

ii. Homomorphic multiplication of plaintexts

An encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts,

$$D(E(m_1,r_1)^{m2} \mod n^2) = m_1 m_2 \mod n,$$
$$D(E(m_2,r_2)^{m1} \mod n^2) = m_1 m_2 \mod n.$$

More generally, an encrypted plaintext raised to a constant *k* will decrypt to the product of the plaintext and the constant,

$$D(E(m_1,r_1)^{k} \mod n^2) = k m_1 \mod n$$

However, given the MREA encryptions of two messages there is no known way to compute an encryption of the product of these messages without knowing the private key.

978-0-7695-4640-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ACCT.2012.74

II. RSA CRYPTOSYSTEM

RSA is based on the principle that some mathematical operations are easier to do in one direction but the inverse is very difficult without some additional information. In case of RSA, the idea is that it is relatively easy to multiply but much more difficult to factor. Multiplication can be computed in polynomial time where as factoring time can grow exponentially proportional to the size of the number.

Key Generation Process:

i. Select two prime numbers p and q.
ii. Find n=p*q, Where n is the modulus that is made public. The length of n is
iii. considered as the RSA key length.
iv. Choose a random number 'e' as a public key in the range 0<e<(p-1)(q-1) such that
v. gcd(e,(p-1)(q-1))=1.
vi. Find private key d such that ed≡1(mod (p-1)(q-1)).

Encryption

i. Consider the device A that needs to send a message to B securely.
ii. Let e be B's public key. Since e is public, A has access to e.
iii. To encrypt the message M, represent the message as an integer in the range 0<M<n.
iv. Cipher text $C = M^e \bmod n$, where n is the modulus.

Decryption

i. Let C be the cipher text received from A.
ii. Calculate Message $M = C^e \bmod n$, where d is B's private key and n is the modulus.

III. OUR SCHEME(MREA)

MREA is an asymmetric-key cryptosystem, meaning that for communication, two keys are required: a public key and a private key. Furthermore, unlike RSA, it is one-way, the public key is used only for encryption, and the private key is used only for decryption. Thus it is unusable for authentication by cryptographic signing. Following is a key generation algorithm for MREA cryptosystem.

A. Key Generation Algorithm:

i. Choose four large prime numbers p, q, r and s randomly and independently of each other. All primes should be of equivalent length.
ii. Compute n = p x q, m= r x s, φ =(p-1) x (q-1) and λ =(r-1) x (s-1).
iii. Choose an integer *e*, $1 < e < \varphi$, such that gcd (e, φ) = 1.
iv. Compute the secret exponent *d*, $1 < d < \varphi$, such that e x d mod φ=1.
v. Select an integer g=m+1.
vi. Compute the modular multiplicative inverse: $\mu = \lambda^{-1} \bmod m$.
vii. The public (encryption) key is (n, m, g, e).
viii. The private (decryption) key is (d, λ, μ).

B. Encryption:

i. Let m be a message to be encrypted where 0< mesg < n.
ii. Select random r where r < m.
iii. Compute ciphertext as: $c=g^{mesg \wedge e \bmod n} \times r^m \bmod m^2$.

C. Decryption

i. Compute message: $m = (((c^{\lambda} \bmod m^2 - 1)/ m) \times \mu \bmod m)^d \bmod n$.

IV. COMPARISON OF RSA AND MREA ALGORITHM

A. Simulation Results The simulation result of the algorithm MREA, implemented in JAVA [4], running on a 2.20 GHz Dual Core Processor and 1 GB RAM has used a1000 characters long message for encryption/decryption. The algorithms (RSA & MREA) have many important parameters affecting its level of security and speed [2]. By increasing the modulus length it is caused of increasing the complexity of decomposing it into its factors. This also increases the length of private key and hence difficulty to detect the key. Another parameter is modular multiplicative inverse μ. Where the modular multiplicative inverse μ is new factor of private key, so it will be more difficult to choose μ by trying all possible private keys (brute force attack) hence the security also increases as well as difficulty of detecting the private key. The RSA and MREA parameters are changed one parameter at a time and the others are kept fixed to study the relative importance. Table 4.1 shows the simulation results of both the algorithms.

Table 4.1: Effect of changing the modulus length m, n and chunk size on the size of two different Private keys, Key generation time, Encryption time and Decryption time MREA cryptosystem, while the size of Public key has kept constant (256 bits, 512 bits).

Size of n,m, d and random no. (bits)	Public key size e (bits)	Chunk Size (bits)	RSA			
			Key Generation time (ms)	Encryption time (ms)	Decryption time (ms)	Total Execution time(ms)
256	256	128	469	109	62	**640**
512	256	128	140	188	218	**546**
512	256	256	140	109	141	**390**
1024	256	128	469	484	1453	**2406**
1024	256	256	469	281	735	**1485**
1024	256	512	469	172	375	**1016**
2048	512	128	2453	2953	15203	**20609**
2048	512	256	2453	1547	5609	**9609**
2048	512	512	2453	812	2750	**6015**
2048	512	1024	2453	515	1375	**4343**

Table 4.2: Effect of changing the modulus length m, n and chunk size on the size of two different Private keys, Key generation time, Encryption time and Decryption time RSA cryptosystem, while the size of Public key has kept constant (256 bits, 512 bits).

Size of n,m, d and random no. (bits)	Public key size (bits)	Chunk Size (bits)	MREA			
			key generation time (ms)	Encryption time (ms)	Decryption time (ms)	Total Execution Time (ms)
256	256	128	484	329	156	**969**
512	256	128	172	1672	968	**2812**
512	256	256	172	890	485	**1547**
1024	256	128	625	11625	6938	**19188**
1024	256	256	625	5860	3515	**10000**
1024	256	512	625	2969	1797	**5391**
2048	512	128	8125	99891	53609	**161625**
2048	512	256	8125	47157	31797	**87079**
2048	512	512	8125	22094	13515	**43734**
2048	512	1024	8125	11219	7016	**26360**

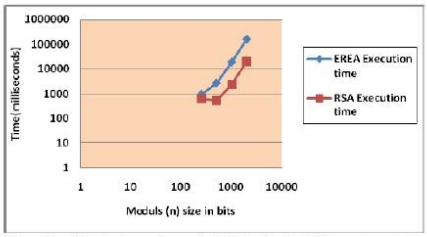

Fig.4.1: Modulus size v/s RSA & MREA algorithm's execution time.

B. Changing the modulus length: Changing the modulus affects the other parameters of the algorithms as shown in Table 4.1. It is clear here that increasing the modulus length (bits) increases the key generation time and encryption\decryption time so the time complexity is increased . Moreover, the length of the secret key (d) increases at the same rate modular multiplicative inverse increases. As a result, increasing the n-bit length provides more security. Hence increasing the n- bit length increases the security but decreases the speed of encryption, decryption and key generation process as illustrated by Figure 4.1 and 4.2.

C. Changing the chunk size: On the basis of simulation results of Table 4.1 & 4.2, following Figure 4.2 shows the effect of chunk size on encryption and decryption time of both the algorithms. Here key generation time of MREA algorithm depends on the size of chunk and as the size of chunk increases key generation time decreases , execution time of MREA algorithm also decreases. RSA algorithm's execution time doesn't depend on modular multiplicative inverse μ .

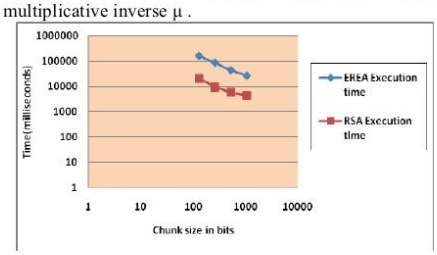

Fig.4.1: Chunk size v/s RSA & MREA algorithm's execution time, taking modulus size 2048 bits.

D. Complexity analysis of MREA cryptosystem

As regards the RSA cryptosystem, it is well-known that the computational complexities for both of the encryption and the decryption are on the order of k^3, where k is the number of bits of the modulus n [7]. The computational complexity of homomorphic problem is *O(k).* So in this way, computational complexity of MREA cryptosystem is on the order of $O(k + k^3)$ i.e. $O(k^3)$ or encryption and on the order of k^3 for decryption. So the computational complexity of MREA is equivalent to RSA cryptosystem. The simulation results of both the algorithms shows that the execution time of MREA is and about 6 times more than RSA.

E. Security analysis of MREA cryptosystem

As the MREA cryptosystem is based on additive homomorphic properties and RSA cryptosystem, additive homomorphic scheme required four prime numbers, it will be more difficult and take long time to factor dual modulus, so one have to factor the dual modulus into its four primes to break the MREA algorithm [7]. If RSA which is based on single modulus, is broken in time x and additive homomorphic based on dual modulus, is broken in time y then the time required to break MREA algorithm is $x*y$. So the security of MREA algorithm is increased as compare to RSA algorithm and it shows that the MREA algorithm is more secure for *Mathematical attacks* [8].

As in MREA double decryption is performed and unlike RSA that is not only based on private key but also based on the subset sum problem so one can't break MREA only guessing the private key only. So it shows that MREA algorithm is more secure as compare to RSA for *Brute force attack* [8].

CONCLUSION

In Public Key cryptography, two different keys are used. One key is used for encryption & the other corresponding key must be used for decryption. No other key can decrypt the message, not even the original (i.e. the first)

key used for encryption. The beauty of this scheme is that every communicating party needs just a key pair for communicating with any number of other communicating parties. It is the first algorithm known to be suitable for signing as well as encryption, The security of the RSA cryptosystem is based on two mathematical problems: the problem of factoring large numbers know mathematical attack and the problem of trying all possible private keys know brute force attack. So MREA improve the security,. MREA is secure as compared to RSA as it is based on the factoring problem as well as decisional composite residuosity assumptions which is the intractability hypothesis.

REFERENCES

[1] Adi Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. *CRYPTO 1982*, pp279–288

[2] Allam Mousa , "Sensitivity of Changing the RSA Parameters on the Complexity and Performance of the Algorithm", *ISSN 1607 – 8926, Journal of Applied Science, Asian Network for Scientific Information*, pages 60-63,2005.

[3] Atul Kahate, "Cryptography and Network Security", *ISBN-10:0-07-064823-9*, Tata McGraw-Hill Publishing Company Limited, India, Second Edition, pages 38-62,152-165,205-240.

[4]. Neal R. Wagner, "The Laws of Cryptography with Java Code", *Technical Report, 2003*, pages 78-112.

[5] Ralph C. Merkle, Martin E. Hellman. "Hiding Information and Signatures in Trapdoor Knapsacks", *IEEE Transactions on Information Theory*, vol. IT-24, 1978, pp. 525-530.

[6] R. Rivest, A. Shamir and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, February 1978, pages 120-126.

[7] RSA Laboratory (2009), "RSA algorithm time complexity", Retrieved from *http://www.rsa.com/rsalabs/node.asp?id=2215* (22 ct. 2009).

[8] William Stallings, "Cryptography and Network Security", *ISBN 81-7758-011-6*, Pearson Education, Third Edition, pages 42-62,121-144,253-297.