Abstract

Cryptography is the study of secure communication techniques. It involves using codes or algorithms to secure information so that it can only be understood by the intended recipient. There are various types of cryptography, including symmetric, asymmetric, and hash functions. Symmetric cryptography involves using the same key to both encrypt and decrypt the information. Asymmetric cryptography, on the other hand, uses different keys for encrypting and decrypting the information.

One popular type of asymmetric cryptography is RSA (Rivest-Shamir-Adleman). This algorithm uses prime numbers to generate a public key and a private key. The public key is used to encode the information, while the private key is used to decode it. RSA is widely used for secure communication, such as in online banking and e-commerce transactions.

In this research paper, I have modified the RSA algorithm by four prime numbers and two public and private keys each. This has increased the complexity and security of the algorithm. Our modified RSA algorithm has been tested and shown to be more secure than the traditional RSA algorithm.

Overall, cryptography is an important field that helps to secure communication and protect sensitive information. By modifying algorithms like RSA, we can continue to improve the security of these techniques and protect ourselves in an increasingly connected world.

Table of Abbreviation

AC	Access Control
AES	Advanced Encryption Standard
CIA	Confidentiality, Integrity, Availability
DES	Data Encryption Standard
GCD	Greatest Common Divisor
IAAA	Identification, Authentication, Authorization, Accountability
NAS	Network Access Server
RADIUS	Remote Authentication Dial-In User Service
RC4	Rivest Cipher 4
RSA	Rivest Shamir Adleman
TACACS+	Terminal Access Controller Access-Control System
WEP	Wired Equivalent Privacy

Table of Contents

Table of Figures

List of Tables

1. Introduction to Cryptographic Systems

1.1. Introduction

In today's digital age, we often exchange sensitive information and secrets online without considering security. To protect this information, modern cryptography uses encryption and decryption to convert it into an unreadable format that can only be accessed by authorized individuals. This is important for maintaining the security of our digital communications and protecting against cyber threats (Gençoğlu, 2019).

Cryptography is a crucial instrument for protecting information that is communicated using computers. It involves converting data into an unreadable format so that only the intended recipient can understand and use it. Cryptography helps to safeguard information from unauthorized access and is used to enable secure communication over the internet. It is accomplished through the use of encryption and decryption keys, with the process of converting plain text into an unreadable format called encryption, and the process of decoding and converting the unreadable text back to readable information using a special digital key called decryption. The main purpose of cryptography is to protect information, such as emails, credit card details, and personal data transmitted over a public network (Gençoğlu, 2019).

There are various types of cryptographic systems, including symmetric-key cryptography, which uses the same key for both encryption and decryption, and asymmetric-key cryptography, which uses a pair of keys for encryption and decryption. Cryptographic systems can also be classified based on their function, such as hash functions, which are used for creating fixed-size hash values for data, and digital signatures, which are used for authenticating the identity of a sender and the integrity of a message.

In this research paper, we will be examining the history and development of cryptographic systems, as well as the different types of cryptographic algorithms and their strengths and weaknesses. We will also be exploring the role of cryptographic systems in information security and the various applications of cryptography in different fields and so on.

1.2. Aim and Objectives

The purpose of this report is to develop and evaluate a new cryptographic algorithm with one widely used cryptographic algorithm as its base and the objective of this report is to:

- Research about information security, CIA triad, Access Control & IAAA framework, Cryptographic system, and different types of cryptographic algorithms.
- Select an existing cryptographic algorithm as a base for creating a new one and conduct in-depth research on it.
- Develop a new cryptographic algorithm to address the flaws of the existing one.
- Test the new cryptographic algorithm with five different cases of variables.
- Evaluate and analyse the new cryptographic algorithm's strengths, weaknesses, and potential applications.

1.3. Key Terminologies

- **Algorithm**: An algorithm is a set of steps or instructions which is used to solve a problem or accomplish a task.
- **Flow chart**: A flowchart is a visual representation of a process or system that shows the steps or decisions involved in the process. It uses symbols and lines to show the flow of information or actions.
- **Encryption**: Encryption is the process of converting plain text into cipher text.
- **Decryption**: Decryption is the process of converting cipher text into plain text.
- **Plain text**: Plain text or message is the original or readable text that works as input into algorithm for encryption.
- **Cipher text**: Cipher text is the raw string of characters which is the output of encryption algorithm.
- **Key**: Key is a series of bits or knowledge used with algorithm as input to create cipher text by manipulating the plain text. Keys are of two types: public key and private key.

- **Public Key**: A public key is a cryptographic key that is used to encrypt data and is made available to the public.

- **Private Key:** A private key is a cryptographic key that is used to decrypt data and is kept secret by the owner.

- **Symmetric-key cryptography**: Symmetric encryption is an encryption method that make use of one key for both encryption and decryption.

- **Asymmetric-key cryptography**: Asymmetric encryption is an encryption method which make use of two keys; public key and private key for encryption and decryption.

2. Information Security

Information security is the practice of protecting data and information from unauthorized access, tampering, and misuse. It involves the use of various policies, rules, and regulations to ensure that sensitive information is kept confidential and secure. Information security is important because it helps to protect sensitive data from being accessed or misused by unauthorized individuals or organizations. It is particularly crucial in the digital age, where large amounts of sensitive data are transmitted and stored online (Michael E. Whitman, 2018).

It is a broad term that encompasses a variety of measures and technologies, including encryption, access controls, firewalls, and antivirus software. These measures are used to protect against a range of threats, including cyber-attacks, data breaches, and other types of unauthorized access. Information security is crucial for businesses, organizations, and individuals to protect their data, assets, and reputation. It is also important for maintaining trust and confidence in the digital world.

Information security is also important because it helps to protect against cyber-attacks and other types of unauthorized access, which can result in the loss or theft of sensitive data, damage to systems and networks, and financial losses. Information security is a critical aspect of modern society, as the reliance on technology and the internet continues to grow. It is essential for ensuring the confidentiality, integrity, and availability of sensitive information and data.

2.1. Components of Information Security

The three critical components which needs to maintain the information security of assets in an enterprise are:

- **Confidentiality:** Confidentiality ensures that data is protected from unauthorized access and only authorized individuals have access to it. This can be achieved through security measures like valid login credentials and access cards. An example is an administrator using a valid username and password to access a computer and preventing unauthorized users from accessing it.

- **Integrity:** Integrity ensures that data remains unchanged as it is transmitted from sender to receiver and no party has interfered with it. This means that the data should not be altered, deleted, or damaged in any way. Measures like encryption, version controls, backups, and user access controls can be used to maintain integrity and prevent tampering, such as when a hacker modifies data and adds malicious payloads or malware.

- **Availability:** Availability ensures that data can be accessed by authorized individuals whenever it is needed. This means that the system and data should be readily available. Measures like backups, disaster recovery, redundancy, and server clustering can be used to maintain availability, such as by having an additional router to route network traffic in case the primary router malfunctions.

Figure 1 Diagram of CIA Triad (Walkowski, 2019).

In order to ensure information security, organizations need to consider all three elements of the CIA triad and implement measures to protect against threats to confidentiality, integrity, and availability. For example, an organization might use encryption to protect confidential data, checksums to ensure the integrity of important documents, and backup systems to ensure the availability of critical information. By addressing all three elements of the CIA triad, organizations can ensure the security of their sensitive information and data.

3. Access Control

Access control is a security measure used in information systems to regulate the access and use of resources, such as data, applications, or networks. It involves defining who is allowed to access certain resources and what actions they are allowed to perform. This is typically done through the use of permissions, roles, and access controls. It uses policies to verify the identity of users and grant appropriate levels of access. Implementing access control is important for web application security because it helps to prevent unauthorized access and data breaches, as well as protect against various types of attacks. Access control is a crucial component of information security because it helps organizations ensure that only the right users have the right level of access to the right resources (Fortinet, 2023).

Access control is important in information security because it helps to ensure that only authorized users have access to resources and that their actions are restricted to their permissions. This helps to protect against unauthorized access, tampering, and misuse of sensitive information.

3.1. Access Control Mechanisms

The main access control mechanisms are:

- **Identification:** This refers to the process of identifying users and verifying their identities. For example, a user might be required to enter a username and password to access a system.
- **Authentication:** This refers to the process of verifying the identity of a user. For example, a system might check a user's credentials, such as a password, to confirm that they are who they claim to be.
- **Authorization:** This refers to the process of granting or denying access to resources based on a user's permissions. For example, a user might be granted access to certain files or applications based on their role in an organization.
- **Accountability:** This refers to the process of tracking and recording user actions and access to resources. For example, a system might keep a log of all user activity, including login and logout times, and the resources accessed.

An example of IAAA in action can be an employee logging into a company's network. First, the employee would identify themselves by entering their username. Next, the system would authenticate the employee by checking their password. If the password is correct, the system would then authorize the employee to access certain resources based on their permissions. Finally, the system would keep a record of the employee's activity, such as the files they accessed and the time they spent on the network.

3.2. Access Control Protocols

The two access control protocols that can be used to implement Authentication, Authorization, Accountability (AAA) services are:

- **RADIUS:** RADIUS is a networking protocol that provides authentication, authorization, and accounting (AAA) services for users on a remote network. It uses a client/server model, with the client being a network access server (NAS) and the server being a RADIUS server. When a user tries to access the network, they send a request to the NAS. The NAS then sends a request for access to the RADIUS server, which responds by either granting access, rejecting the request, or requesting more information. RADIUS also encrypts all AAA data packets to provide an extra level of security (Fortinet, 2023).
- **TACACS+:** Like RADIUS, TACACS+ is a networking protocol that uses a client/server model to connect users. It allows more control over the authorization of commands and separates the authentication and authorization processes, unlike RADIUS which combines them. TACACS+ works by requiring a secret key that is shared by the client and the TACACS+ system. If a valid key is presented, the connection is allowed to proceed. TACACS+ also encrypts its AAA packets for added security (Fortinet, 2023).

4. Cryptography

Cryptography is the practice of secure communication between parties, allowing only the sender and receiver to view the message. It involves the encryption of ordinary text, known as plaintext, into unreadable text called ciphertext. Cryptography can also be used to obscure information within images through techniques like microdots and merging. If the transmission or storage medium is compromised, the transmitted ciphertext is practically useless to unauthorized individuals without the use of a valid key for decryption. Cryptography can be related to the process of encrypting and decrypting messages to protect their confidentiality (Kaspersky, 2023).

Overall, the main purpose of cryptography is to protect sensitive information from being accessed or understood by unauthorized individuals. It helps to ensure the confidentiality and integrity of transmitted data, ensuring that only the intended parties are able to access and understand the information. In the digital age, cryptography plays a vital role in protecting data and communication from cyber threats and attacks. It is used in various industries and applications, including online banking, e-commerce, and government communication.

4.1. History of Cryptography

Cryptography involves the use of algorithms to protect information as it is transmitted from sender to receiver. The term "cryptography" comes from the Greek words "kryptos" meaning "hidden" and "graphein" meaning "writing". It involves encoding plain text into cipher text using a key to make the original message unreadable to unauthorized individuals and ensure confidentiality and integrity in information. Cryptography involves the use of algorithms to protect information as it is transmitted from sender to receiver. The term "cryptography" comes from the Greek words "kryptos" meaning "hidden" and "graphein" meaning "writing". It involves encoding plain text into cipher text using a key to make the original message unreadable to unauthorized individuals and ensure confidentiality and integrity in information (Michael E. Whitman, 2018).

Cryptography has been used for centuries to protect information as it is transmitted from sender to receiver. The use of cryptography originated when writing was invented, as

people began to communicate secretly due to the development of power, politics, and battles. Egypt and Rome are considered the birthplaces of cryptography, with the development of written hieroglyphs in Egypt around 4000 years ago and mono-alphabetic substitution in Rome around 500-600 BC (tutorialspoint, 2022).

Figure 2 Hieroglyph − The Oldest Cryptographic Technique (tutorialspoint, 2022).

During the European Renaissance, cryptography techniques advanced significantly. The 15th century saw the development of Vigenère coding, a poly-alphabetic substitution, and the 20th century saw the invention of mechanical and electromechanical cipher machines like the Enigma and Typex rotor machines. During World War 2, cryptography and cryptoanalysis became more mathematical in nature (tutorialspoint, 2022).

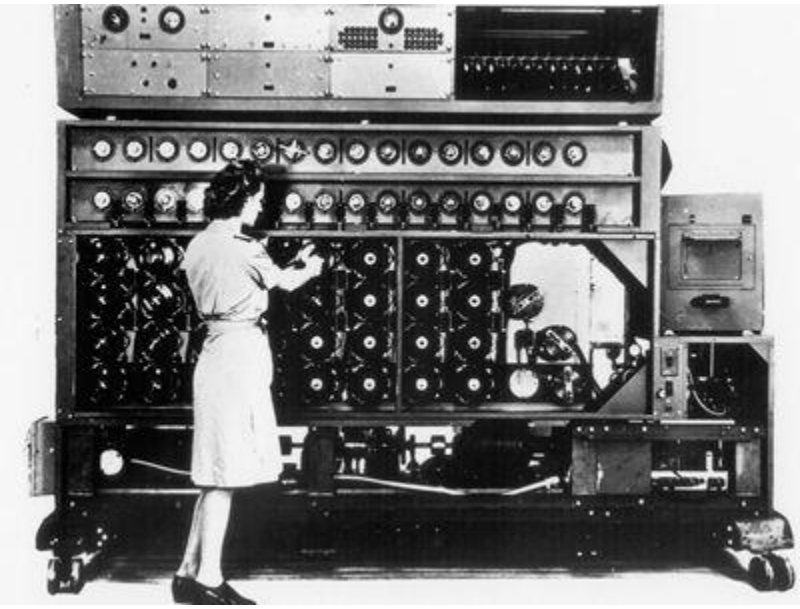

Figure 3 Enigma Machine used in WW II (Copeland, 2019).

The modern era of cryptography began in the 1980s, with the invention of the one-time pad and the development of stream and block symmetric encryption systems like RC4, WEP, Blowfish, DES, and AES. AES was even adopted by the US government for encrypting sensitive information. The digital era of the 1970s led to the development of

asymmetric encryption systems like RSA and Diffie-Hellman, which use two keys for encryption and decryption (Wong, 2021).

Overall, the history of cryptography has seen the development of a wide range of cryptographic algorithms and techniques, with a focus on security and the ability to encode and decode sensitive information.

4.2. Basic Principles of Cryptography

- **Confidentiality:** Cryptography ensures that the confidentiality of sensitive information is protected by encoding it in a way that makes it unreadable to unauthorized individuals.
- **Data Integrity:** Cryptography ensures the integrity of information by detecting and preventing tampering or modification.
- **Authentication:** Cryptography helps authenticate the identity of a sender and the integrity of a message through the use of digital signatures.
- **Authorization:** Cryptography helps protect against cyber-attacks and other types of unauthorized access.
- **Non-repudiation:** Cryptography ensures that the sender of a message cannot later deny sending the message.

4.3. Types of Cryptography

There are two main types of cryptography: symmetric-key cryptography and asymmetric-key cryptography. Both of types of cryptography have been explained below.

4.3.1. Symmetric-key Cryptography

Symmetric-key cryptography involves the use of the same key for both encryption and decryption. This means that both the sender and receiver must have a copy of the key in order to communicate securely. Symmetric key algorithms can be implemented as either block ciphers or stream ciphers. Block ciphers encrypt input in blocks of plaintext, while stream ciphers encrypt individual characters. Symmetric-key cryptography is faster than asymmetric cryptography, which uses a pair of keys for encryption and decryption (Shivani Sharma, 2017).

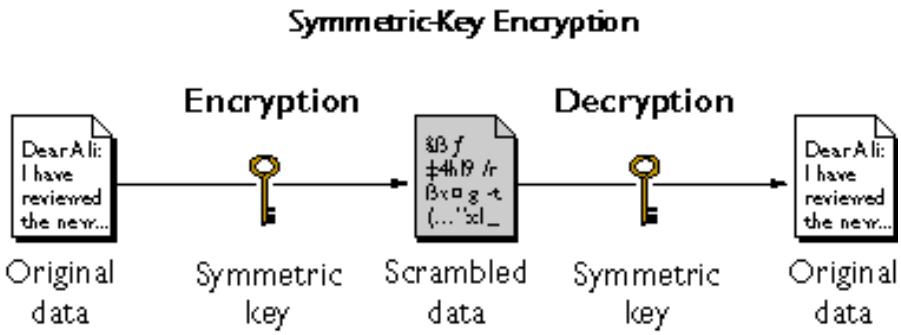

Figure 4 Diagram of how Symmetric-Key works (IBM, 2021).

An example of symmetric cryptography is when Max and Ruby want to communicate securely over the internet. Max and Ruby both agree on a secret key and share it with each other. Max uses the secret key to encrypt a message and sends it to Ruby. Ruby then uses the same secret key to decrypt the message and read it. This process is repeated for all future communications between Max and Ruby.

Advantages of Symmetric-Key Cryptography:

- **Speed and efficiency**: Symmetric-key cryptography is relatively simple and requires less computational power, making it suitable for encrypting large amounts of data quickly.
- **Ease of implementation and management**: Both parties can easily agree on a key beforehand and use it for all their communication, making the process more convenient.
- **Security**: Symmetric-key cryptography is generally more secure than other types of encryption, as the key is not transmitted over the internet and is therefore less likely to be compromised.

4.3.2. Asymmetric-key Cryptography

Asymmetric-key cryptography involves the use of two different keys for encryption and decryption. One key, known as the public key, is used for encryption and is made publicly available. The other key, known as the private key, is used for decryption and is kept secret. This type of cryptography provides more stability than symmetric systems, which use the same key for both encryption and decryption. Asymmetric-key cryptography is commonly used for secure communication and authentication, as it allows for the exchange of secure messages without the need to share a secret key.

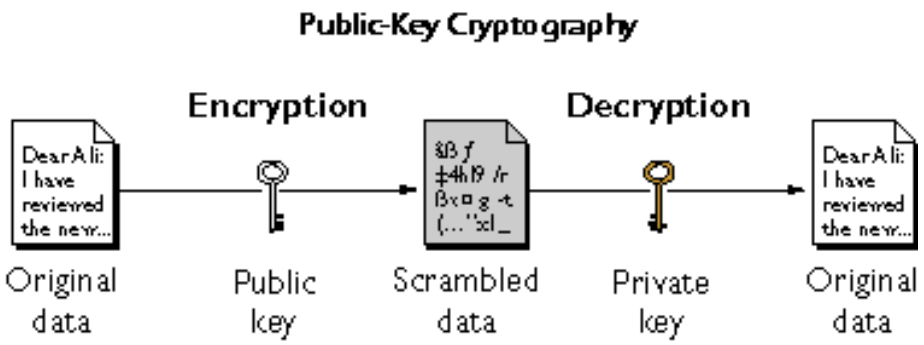

Figure 5 Diagram of how Asymmetric-key works (IBM, 2021).

An example of asymmetric cryptography is when Max wants to send a secure message to Ruby. Max uses Ruby's public key, which is available to anyone, to encrypt the message. Ruby then uses her private key, which only she has access to, to decrypt the message. This process ensures that only Ruby can read the message, as her private key is needed to decrypt it. Max can also use his private key to sign the message, authenticating his identity and the integrity of the message.

Advantages of Asymmetric-Key Cryptography:

- **Convenience**: Because the sender and receiver use different keys, there is no need to exchange keys beforehand in order to establish secure communication. This makes asymmetric-key cryptography suitable for establishing secure communication with multiple parties.
- **Security**: The private key, which is used for decrypting the message, is not shared with anyone and is therefore less likely to be compromised. As a result, asymmetric-key cryptography is generally more secure than symmetric-key cryptography.
- **Non-repudiation**: Asymmetric-key cryptography allows for non-repudiation, meaning that the sender of the message cannot deny sending it. This makes it suitable for applications such as secure email, digital signatures, and secure file transfers.

5. Background of Rivest Shamir Adleman (RSA)

RSA is a method for encrypting and decrypting messages that was developed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. It is a type of public-key cryptography, which was first introduced in 1976 by Whitfield Diffie, Martin Hellman, and Ralph Merkle. Rivest, Shamir, and Adleman developed RSA by trying to create an unbreakable key system and then testing it to make sure it was secure. They were successful after trying 42 different key systems (Adleman, 2018).

The Rivest-Shamir-Adleman (RSA) encryption algorithm is an asymmetric algorithm, which means that it uses a pair of linked keys to encrypt and decrypt data. These keys are known as the private and public keys, with the private key being kept secret by the key pair creator and the public key being accessible to anyone. One of the unique features of RSA is that either the private or public key can be used to encrypt the data, while the other key is used to decrypt it. This means that the private key can be kept secure, while the public key can be shared with others without compromising the security of the encrypted data. RSA is considered to be the most widely used asymmetric encryption algorithm due to its effectiveness and widespread adoption in many products and services (Encryption Consulting LLC, 2023).

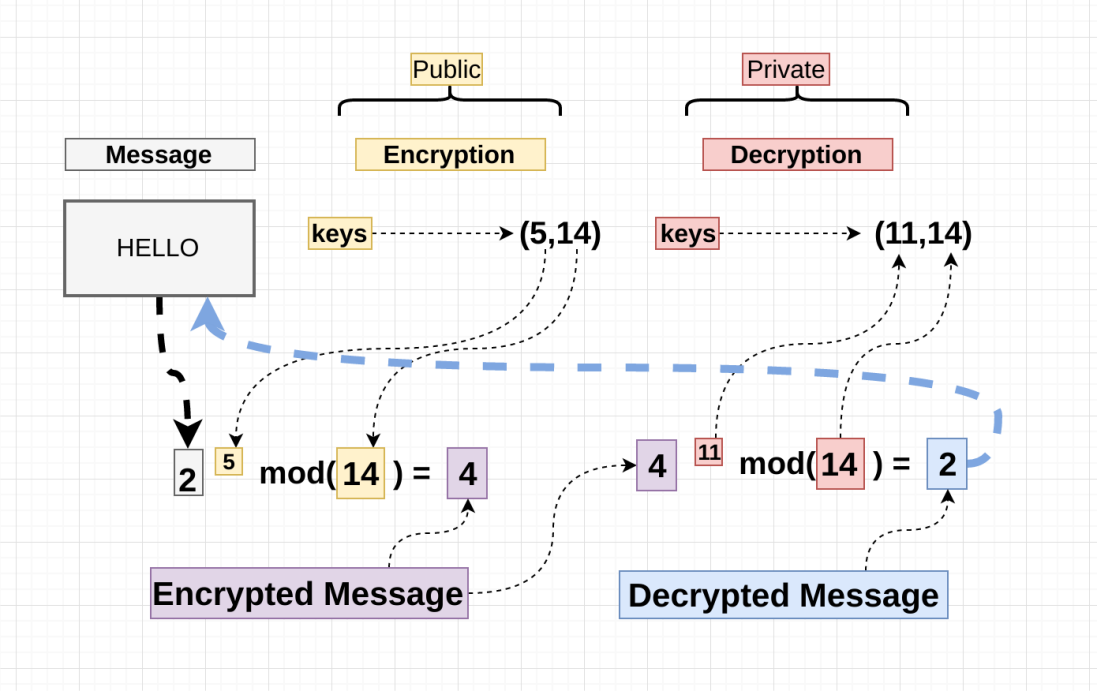

Figure 6 Example of how RSA works (keyfactor, 2021).

In the example shown in the image, the plaintext "2" is encrypted using the RSA algorithm. To do this, the plaintext is raised to the power of the public key and then divided by a public modulus. The result of this calculation, known as the ciphertext, is sent to the recipient. At the other end, the same calculation is performed using the private key instead of the public key to decrypt the ciphertext and produce the original plaintext. The keys are chosen in a way that allows the decryption process to reverse the encryption process, making it possible to securely transmit the data (keyfactor, 2021).

5.1. Key Generation

The security and complexity of the RSA algorithm comes from the generation of the public and private keys. The key generation steps are carried out in following steps:

- Choose two prime numbers p and q such that p != q [which means that p should not be equal to q and vice versa].
- Calculate the n where n = p x q [n is the product of p and q].
- Calculate the Ø(n) where Ø(n) = (p - 1) x (q - 1).
  Here, Ø is the symbol for phi.
- Choose the public exponent e such that GCD (e, Ø(n)) = 1 and 1 < e < n
  Here, GCD is abbreviation of Greatest Common Divisor.
- Choose the private exponent d such that 1 < d < Ø(n) and e x d mod Ø(n) = 1.
- After that, public key (e, n) and private key (d, n) is obtained.

After the generation of keys, the message (m) can be encrypted by applying:

Cipher text (c) = $(m^e \ mod \ n)$

And the cipher text (c) can be decrypted by applying:

Message (m) = $\left(c^d \ mod \ n\right)$

5.2. Encryption and Decryption

Let's choose the prime numbers to be p = 11 and q = 17. Here, p is not equal to q. Let A be our plain text / message. For the calculation, we will need to have integer value. Hence, referring to the ASCII table, A is converted to 65.

Keys Generation:

The step for generating the keys is as follows.

1. n = p x q
   = 11 x 17
   = 187

2. Ø(n) = (p - 1) x (q - 1)
   = (11 - 1) x (17 - 1)
   = 10 x 16
   = 160

3. e = 7 since GCD(e, Ø(n)) = 1 and 1 < e < Ø(n).
   [i.e., GCD(7, 160 = 1) and 1 < 7 < 160]

4. d = 23 since e x d mod Ø(n) = 1
   [i.e., 7 x 23 mod 160 = 1) and 1 < 23 < 160]

5. Hence, we get the keys: public key pair (e, n) = (7, 187) and private key pair (d, n) = (23, 187)

Encryption of the message:

The message 'A', 65 is encrypted as following.

Cipher text (c) = $(m^e \ mod \ n)$
= $(65^7 \ mod \ 187)$
= 142

Decryption of the cipher text:

The cipher text 142 can be decrypted as following.

Message (m) = $(c^d \ mod \ n)$
= $(142^{23} \ mod \ 187)$
= 65

Refer to this [python code] for encryption and decryption that uses the above values.

5.3. Advantages and Disadvantages

The advantages of using RSA algorithm are:

- It is resistant to attacks based on quantum computers, which makes it a good choice for long-term data protection.
- It can be used for both encryption and digital signatures, which makes it a versatile tool for secure communication.
- It can be implemented relatively quickly.
- It is secure and reliable for sending private information.
- Distributing the public key to consumers is simple (Kumari, 2022).

The disadvantages of using RSA algorithm are:

- RSA uses asymmetric encryption, which means that it uses a pair of linked keys to encrypt and decrypt data.
- The key pairs with small size can be easily brute forced.
- To fully encrypt data, both symmetric and asymmetric encryption are often used. In some cases, RSA may not be sufficient on its own.
- It may be necessary to use a third party to confirm the validity of public keys.
- Decrypting data with RSA requires a lot of processing power on the receiver's end.
- RSA is not suitable for encrypting public data, such as in electoral voting (Kumari, 2022).

5.4. Approach of attacking RSA

There are several ways that attackers might try to break the RSA algorithm. Some of them are listed below.

Brute force attack:

In this type of attack, the attacker tries all possible keys until they find the correct one. For example, an attacker might try every possible combination of letters and numbers until they find the correct decryption key for a message.

Mathematical attacks:

There are a number of mathematical techniques that can be used to try and break RSA, such as factoring the modulus used in the key generation process. For example, an

attacker might try to factor the large prime numbers used in an RSA key in order to figure out the value of the key.

Side-channel attacks:

These types of attacks try to gather information about the encryption or decryption process by observing things like the amount of time it takes to perform the calculation, or the amount of power used. For example, an attacker might try to measure the amount of time it takes to decrypt a message in order to infer the value of the key.

Physical attacks:

In some cases, an attacker might try to gain physical access to a device that is performing RSA encryption or decryption in order to extract the key or other sensitive information. For example, an attacker might try to steal a laptop that has an RSA key stored on it.

Social engineering:

Attackers might try to trick users into revealing their keys or other sensitive information through various social engineering techniques, such as phishing scams or pretexting. For example, an attacker might send an email pretending to be from a trusted source and asking the user to reveal their RSA key.

6. Development of Modified Rivest Shamir Adleman (RSA)

I have modified the RSA algorithm to further enhance its security and complexity. In my modified version, I have made the use of four prime numbers to generate the public and private keys, and n is the product of these four numbers. Phi is also calculated from the four prime numbers. This makes it much more difficult for an attacker to break the key using a brute force attack or other methods, because they would have to factor a much larger number in order to determine the value of the keys. In addition, I have also modified the encryption and decryption process which is based on double encryption of the plain text and double decryption of the cipher text. This further increases the security of the system because it would be virtually impossible to use a brute force attack to determine the value of both pairs of keys. The keys generation, encryption and decryption process have been listed in following steps:

- Choose four prime numbers p, q, r, s such that p != q != r != s [which means that p, q, r, s should not be equal to each other].
- Calculate the n where n = p x q x r x s [n is the product of p, q, r, s].
- Calculate the Ø(n) where Ø(n) = (p - 1) x (q - 1) x (r - 1) x (s - 1).
  Here, Ø is the symbol for phi.
- Choose two random prime numbers for first public exponent e1 such that GCD (e1, Ø(n)) = 1 and 1 < e1 < n and second public exponent e2 such that GCD (e2, Ø(n)) = 1 and 1 < e2 < n.
  Here, GCD is abbreviation of Greatest Common Divisor.
- Calculate the two private exponents where d1 should qualify 1 < d1 < Ø(n) and e1 x d1 mod Ø(n) = 1 and d2 should qualify 1 < d2 < Ø(n) and e2 x d2 mod Ø(n) = 1.
- After that, the two public key pair (e1, n) and (e2, n) is obtained and two private key pair (d1, n) and (d2, n) is also obtained.

To encrypt the message, Use:

Cipher Text (c) = $(m^{e1} \bmod n)^{e2} \bmod n$.

To decrypt the cipher text, Use:

Message (m) = $\left(c^{d2} \bmod n\right)^{d1} \bmod n$.

6.1. Key Generation Algorithm

The algorithm that can be used for the key generation for our modified RSA as follows.

Step 1: Start

Step 2: Select four random prime numbers p, q, r, and s.

Step 3: Calculate n by multiplying p, q, r, and s.

Step 4: Calculate phi by multiplying p - 1, q - 1, r - 1, and s - 1 [phi can be represented as Ø].

Step 5: Select two random prime numbers e1 and e2 such that e1 and e2 are greater than 1 and lesser than phi.

Step 6: If GCD of e1 and phi is not equal to 1 and GCD of e2 and phi is not equal to 1. Go back to Step 5.

Step 7: Select d1 and d2 such that d1 and d2 are greater than 1 and lesser than phi.

Step 8: If e1 multiplied by d1 mod phi is not equal to 1 and e2 multiplied by d2 mod phi is not equal to 1. Go back to Step 7.

Step 9: After that, public key pairs (e1, n) and (e2, n) are obtained and private key pair (d1, n) and (d2, n) are also obtained.

Step 10: Public key is given to the parties who needs to send the encrypted message and private keys are placed safely and used while decrypting the cipher.

Step 11: Stop

The algorithm that can be used for the encryption and decryption for our modified RSA as follows.

6.2. Encryption Algorithm

Step 1: Start

Step 2: Take message (m), and public keys e1, e2 and n.

Step 3: Calculate cipher text (c) = $(m^{e1} \ mod \ n \ )^{e2} \ mod \ n$.

[(m to the power e1 mod by n) to the power e2 mod by n].

Step 4: Cipher text is obtained.

Step 5: Stop

6.3. Decryption Algorithm

Step 1: Start

Step 2: Take cipher text c, and private keys d1, d2 and n.

Step 3: Calculate message (m) = $(c^{d2}\ mod\ n\ )^{d1}\ mod\ n$.

[(c to the power d2 mod by n) to the power d1 mod by n].

Step 4: Plain Text / Original Message is obtained.

Step 5: Stop

6.4. Flow Chart

The flow chart for key generation of our modified RSA may be represented as the image below.

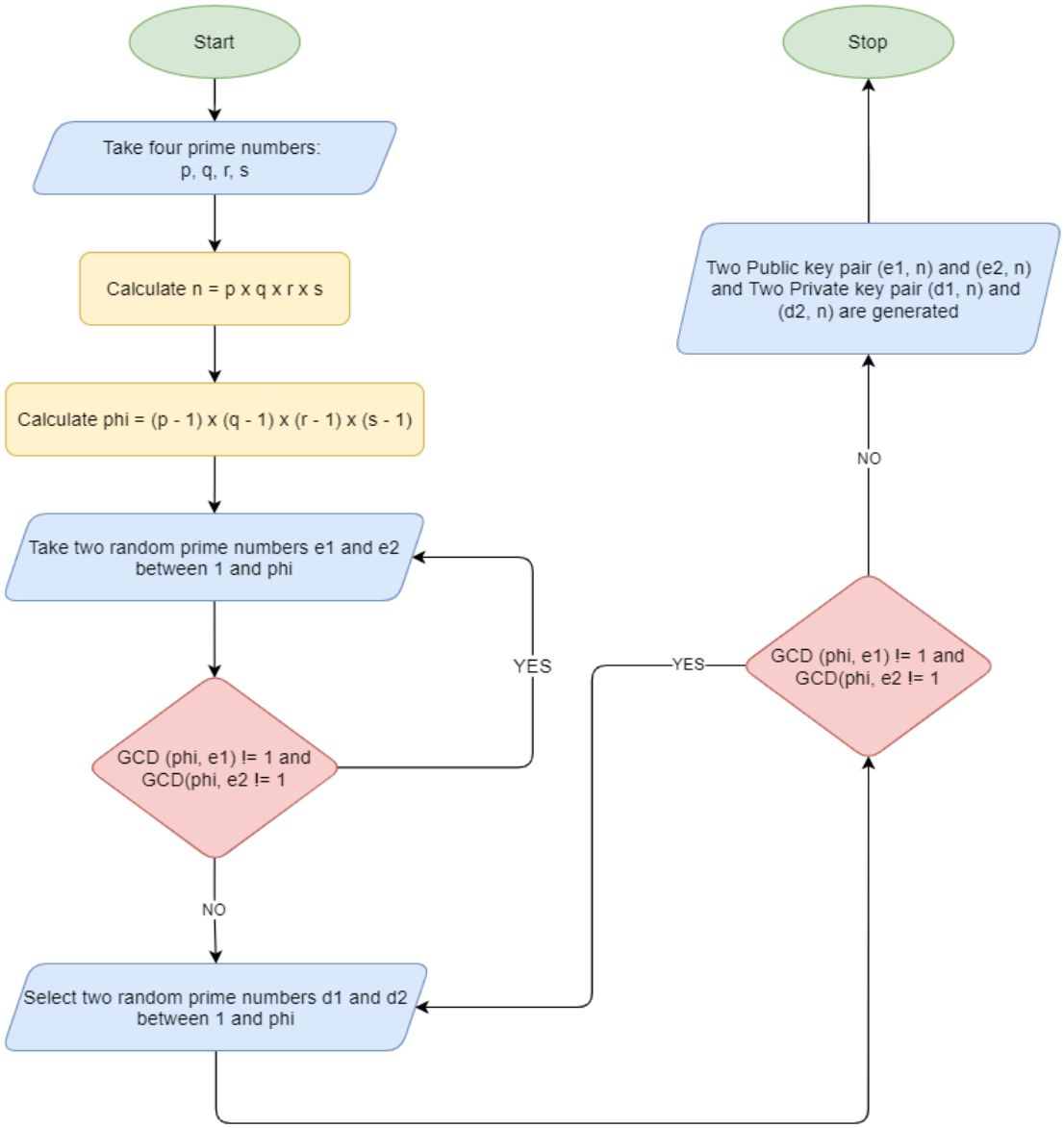

Figure 7 Flowchart for Key Generation of Modified RSA

The flow chart for the encryption and decryption process for our modified RSA may be represented as the image below.

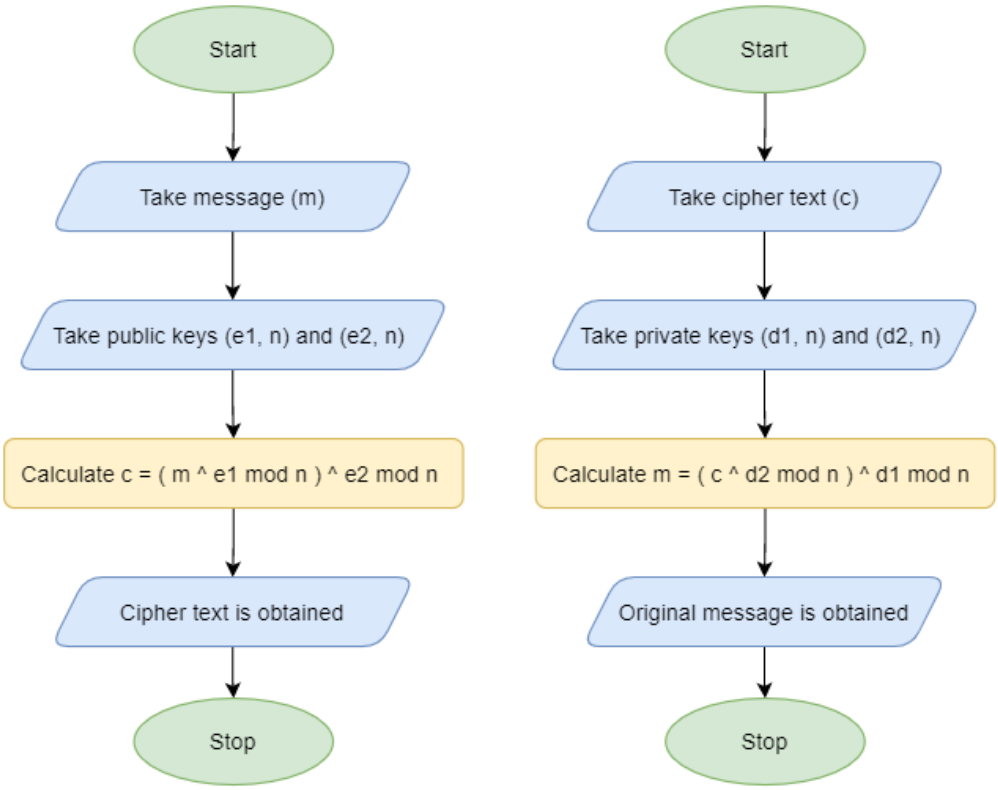

Figure 8 Flowchart for Encryption and Decryption of Modified RSA

7. Testing

Below are the five test cases that I have done for the proposed RSA algorithm.

7.1. Test 1

Let p, q, r, s = 3, 5, 7, 11

Message to encrypt = ‘A’

Convert A into decimal as per ASCII we get, Message (m) = 65

n = p x q x r x s

= 3 x 5 x 7 x 11

= 1155

Ø(n) = (p - 1) x (q - 1) x (r - 1) x (s - 1)

= 2 x 4 x 6 x 10

= 480

Choosing e1 = 7 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 7 < 480 and GCD (7, 480) = 1

Choosing e2 = 11 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 11 < 480 and GCD (11, 480) = 1

Choosing d1 = 343 to qualify 1 < d1 < Ø(n) and e1 x d1 mod Ø(n) = 1

Since, 1 < 343 < 480 and 7 x 343 mod 480 = 1

Choosing d2 = 131 to qualify 1 < d2 < Ø(n) and e2 x d2 mod Ø(n) = 1

Since, 1 < 131 < 480 and 11 x 131 mod 480 = 1

Encryption	Decryption
c = $(m^{e1} \ mod \ n)^{e2} \ mod \ n$	m = $(c^{d2} \ mod \ n)^{d1} \ mod \ n$
c = $(65^{7} \ mod \ 1155)^{11} \ mod \ 1155$	m = $(725^{131} \ mod \ 1155)^{343} \ mod \ 1155$
c = 725	m = 65

Table 1 Encryption, Decryption and Result for Test 1

7.2. Test 2

Let p, q, r, s = 3, 5, 7, 11

Message to encrypt = ‘Z’

Convert Z into decimal as per ASCII we get, Message (m) = 90

n = p x q x r x s

= 3 x 5 x 7 x 11

= 1155

Ø(n) = (p - 1) x (q - 1) x (r - 1) x (s - 1)

= 2 x 4 x 6 x 10

= 480

Choosing e1 = 7 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 7 < 480 and GCD (7, 480) = 1

Choosing e2 = 11 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 11 < 480 and GCD (11, 480) = 1

Choosing d1 = 343 to qualify 1 < d1 < Ø(n) and e1 x d1 mod Ø(n) = 1

Since, 1 < 343 < 480 and 7 x 343 mod 480 = 1

Choosing d2 = 131 to qualify 1 < d2 < Ø(n) and e2 x d2 mod Ø(n) = 1

Since, 1 < 131 < 480 and 11 x 131 mod 480 = 1

Encryption	Decryption
c = $(m^{e1} \ mod \ n)^{e2} \ mod \ n$	m = $(c^{d2} \ mod \ n)^{d1} \ mod \ n$
c = $(90^{7} \ mod \ 1155)^{11} \ mod \ 1155$	m = $(1140^{131} \ mod \ 1155)^{343} \ mod \ 1155$
c = 1140	m = 90

Table 2 Encryption, Decryption and Result for Test 2

7.3. Test 3

Let p, q, r, s = 3, 5, 7, 11

Message (m) = 999

n = p x q x r x s

= 3 x 5 x 7 x 11

= 1155

Ø(n) = (p - 1) x (q - 1) x (r - 1) x (s - 1)

= 2 x 4 x 6 x 10

= 480

Choosing e1 = 109 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 109 < 480 and GCD (109, 480) = 1

Choosing e2 = 479 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 479 < 480 and GCD (479, 480) = 1

Choosing d1 = 229 to qualify 1 < d1 < Ø(n) and e1 x d1 mod Ø(n) = 1

Since, 1 < 229 < 480 and 109 x 229 mod 480 = 1

Choosing d2 = 479 to qualify 1 < d2 < Ø(n) and e2 x d2 mod Ø(n) = 1

Since, 1 < 479 < 480 and 479 x 479 mod 480 = 1

Encryption	Decryption
c = $(m^{e1} \ mod \ n \ )^{e2} \ mod \ n$	m = $(c^{d2} \ mod \ n \ )^{d1} \ mod \ n$
c = $(999^{109} \ mod \ 1155 \ )^{479} \ mod \ 1155$	m = $(339^{479} \ mod \ 1155 \ )^{229} \ mod \ 1155$
c = 339	m = 999

Table 3 Encryption, Decryption and Result for Test 3

7.4. Test 4

Let p, q, r, s = 11, 13, 17, 19

Message to encrypt = ‘F’

Convert F into decimal as per ASCII we get, Message (m) = 70

n = p x q x r x s

= 11 x 13 x 17 x 19

= 46189

Ø(n) = (p - 1) x (q - 1) x (r - 1) x (s - 1)

= 10 x 12 x 16 x 18

= 34560

Choosing e1 = 7 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 7 < 34560 and GCD (7, 34560) = 1

Choosing e2 = 11 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 11 < 34560 and GCD (11, 34560) = 1

Choosing d1 = 29623 to qualify 1 < d1 < Ø(n) and e1 x d1 mod Ø(n) = 1

Since, 1 < 29623 < 34560 and 7 x 29623 mod 34560 = 1

Choosing d2 = 18851 to qualify 1 < d2 < Ø(n) and e2 x d2 mod Ø(n) = 1

Since, 1 < 18851 < 34560 and 11 x 18851 mod 34560 = 1

Encryption	Decryption
c = $(m^{e1} \ mod \ n)^{e2} \ mod \ n$	m = $(c^{d2} \ mod \ n)^{d1} \ mod \ n$
c = $(70^{7} \ mod \ 46189)^{11} \ mod \ 46189$	m = $(46051^{18851} \ mod \ 46189)^{29623} \ mod \ 46189$
c = 46051	m = 70

Table 4 Encryption, Decryption and Result for Test 4

7.5. Test 5

Let p, q, r, s = 13, 17, 19, 23

Message to encrypt = ‘X’

Convert X into decimal as per ASCII we get, Message (m) = 88

n = p x q x r x s

= 13 x 17 x 19 x 23

= 96577

Ø(n) = (p - 1) x (q - 1) x (r - 1) x (s - 1)

= 12 x 16 x 18 x 22

= 76032

Choosing e1 = 13 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 13 < 76032 and GCD (13, 76032) = 1

Choosing e2 = 19 to qualify 1 < e1 < Ø(n) and GCD (e1, Ø(n)) = 1

Since, 1 < 19 < 76032 and GCD (19, 76032) = 1

Choosing d1 = 46789 to qualify 1 < d1 < Ø(n) and e1 x d1 mod Ø(n) = 1

Since, 1 < 46789 < 76032 and 13 x 46789 mod 76032= 1

Choosing d2 = 64027 to qualify 1 < d2 < Ø(n) and e2 x d2 mod Ø(n) = 1

Since, 1 < 64027 < 76032 and 19 x 64027 mod 76032= 1

Encryption	Decryption
c = $(m^{e1} \ mod \ n)^{e2} \ mod \ n$	m = $(c^{d2} \ mod \ n)^{d1} \ mod \ n$
c = $(88^{13} \ mod \ 96577)^{19} \ mod \ 96577$	m = $(25035^{64027} \ mod \ 96577)^{46789} \ mod \ 96577$
c = $(80116)^{19} \ mod \ 96577$	m = $(80116)^{46789} \ mod \ 96577$
c = 25035	m = 88

Table 5 Encryption, Decryption and Result for Test 5

Refer to this simple python code that is set to the values in this test to verify the results.

8. Evaluation

The modified RSA algorithm uses four prime numbers for key generation, instead of the two used in the conventional RSA algorithm. This increases the security of the system, but also significantly increases the time required for encryption, decryption, and key generation. However, the main goal of the modified algorithm is to increase security, so the trade-off in terms of increased time is acceptable.

Testing has shown that the security of the modified RSA algorithm is increased due to the use of two key pairs for encryption and decryption. If an attacker tries to use a brute force attack to determine the keys, the process will be much more time-consuming due to the use of four large prime numbers. In addition, even if an attacker is able to determine one of the keys, they will not be able to decrypt the message without the other key. Overall, the modified RSA algorithm provides improved security compared to the conventional RSA algorithm, especially against brute force attacks on the keys. The modified RSA algorithm provides a good balance of security and practicality, making it a strong choice for protecting sensitive data.

8.1. Advantages of proposed algorithm

- **Improved security:** The use of four prime numbers and double key pairs makes it more difficult for an attacker to determine the value of the keys, which increases the overall security of the algorithm.
- **Resistance to brute force attacks:** The use of four large prime numbers makes it much more time-consuming for an attacker to use a brute force attack to try and determine the keys.
- **Additional layer of protection:** Even if an attacker is able to determine one of the keys, they will not be able to decrypt the message without the other key, providing an additional layer of protection.
- **Versatility:** The modified algorithm can be used for both encryption and digital signatures, making it a versatile tool for secure communication.
- **Compatibility with conventional RSA:** The modified algorithm is based on the conventional RSA algorithm, which means that it is compatible with existing systems and infrastructure that use RSA.

8.2. Weakness of proposed algorithm

Below are some of the weaknesses of the proposed algorithm.

- **Increased computation time:** The use of four prime numbers and double key pairs may result in longer computation times for key generation, message encryption, and decryption, which could make the algorithm less practical in some situations.
- **Resource-intensive:** The modified algorithm may require a lot of computing resources to perform the necessary calculations, which could be a drawback in some cases.
- **Sensitivity to errors:** A slight mistake in the calculations could result in a large difference in the final result, which could compromise the security of the algorithm.
- **Risk of key loss:** If one of the private keys is lost, it may be almost impossible to decrypt the ciphertext, which could compromise the security of the system.

8.3. Application area of proposed algorithm

The modified algorithm could potentially be used in any situation where the conventional RSA algorithm is used, as long as its additional complexity and resource requirements are acceptable. This could include applications such as secure communication, data encryption, and digital signatures. It is worth noting that the modified algorithm may be more suitable for certain types of applications than others, depending on the specific requirements and constraints of those applications. For example, the modified algorithm may be more suitable for applications that require a high level of security, even if it comes at the cost of increased computation time and resource usage.

9. Conclusion

In conclusion, the topic of information security is of great importance in today's digital age. With the increasing reliance on the internet and the vast amount of sensitive information being transmitted and stored online, the security threats have also evolved. It is therefore crucial to protect sensitive data and information from unauthorized access, tampering, and misuse.

The history of cryptography dates back to ancient civilizations, and it has played a vital role in securing information and protecting against security threats. There are various types of cryptographic algorithms, each with its own strengths and weaknesses. Throughout this coursework, I got the opportunity to learn about various cryptographic algorithm and also about RSA, a widely used asymmetric cryptographic algorithm which is widely used for secure data transmission and authentication.

Based on my research and analysis, I decided to choose RSA and develop a new cryptographic algorithm by modifying its some properties. I carried out in-depth research on RSA and identified its flaws, and then developed a new algorithm that addresses these weaknesses. I tested my new algorithm with five different cases of variables and evaluated its strength and potential applications.

Overall, this coursework project has been a valuable learning experience, as I was able to learn about the history of cryptography, different types of cryptographic algorithms, and the importance of information security. I was also able to apply this knowledge to develop a new cryptographic algorithm by modifying RSA, demonstrating the practical applications of my research.

10. References

Adleman, L. (2018). *RSA Cryptography*. Retrieved January 7, 2023, from National Inventors Hall of Fame: https://www.invent.org/inductees/leonard-adleman

Copeland, B. (2019, April 4). Ultra. Retrieved from https://www.britannica.com/topic/Ultra-Allied-intelligence-project

Encryption Consulting LLC. (2023). *What is RSA? How does an RSA work?* Retrieved January 7, 2023, from https://www.encryptionconsulting.com/education-center/what-is-rsa/

Fortinet. (2023). *AAA Security*. Retrieved January 6, 2023, from https://www.fortinet.com/resources/cyberglossary/aaa-security

Fortinet. (2023). *What Is Access Control?* Retrieved January 2023, from https://www.fortinet.com/resources/cyberglossary/access-control

Gençoğlu, M. T. (2019). *Importance of Cryptography in Information Security, 21*(1). Retrieved January 2023

IBM. (2021, March 1). *Public key cryptography*. Retrieved from https://www.ibm.com/docs/en/ztpf/2020?topic=concepts-public-key-cryptography

IBM. (2021, March 01). *Symmetric cryptography*. Retrieved from https://www.ibm.com/docs/en/ztpf/2020?topic=concepts-symmetric-cryptography

Kaspersky. (2023). *What is Cryptography?* Retrieved January 6, 2023, from https://www.kaspersky.com/resource-center/definitions/what-is-cryptography

keyfactor. (2021, August 08). *Types of Encryption Algorithms + Pros and Cons for Each*. Retrieved January 07, 2023, from https://www.keyfactor.com/resources/types-of-encryption-algorithmns/

Kumari, M. (2022, October 13). *RSA Full Form - History, Advantages and Disadvantages*. Retrieved from BYJU'S Exam Prep: https://byjusexamprep.com/rsa-full-form-i

Michael E. Whitman, H. J. (2018). *Principles of Information Security.* Cengage Learning.

Shivani Sharma, Y. G. (2017). *Study on Cryptography and Techniques.* Noida, Uttar Pradesh: IJSRCSEIT.

tutorialspoint. (2022). *Origin of Cryptography*. Retrieved January 5, 2023, from https://www.tutorialspoint.com/cryptography/origin_of_cryptography.htm

Walkowski, D. (2019, July 08). *What Is the CIA Triad?* Retrieved from F5: https://www.f5.com/labs/learning-center/what-is-the-cia-triad

Wong, D. (2021). *Real-World Cryptography.* Manning Publications.

11. Bibliography

Abderrahmane Nitaj, T. R. (2022). *Factoring RSA moduli with weak prime factors.*

CTF 101. (2021). *What is RSA ?* Retrieved from https://ctf101.org/cryptography/what-is-rsa/

Kotas, W. A. (2000). *A Brief Hist A Brief History of Cr y of Cryptography.* TRACE: Tennessee Research and Creative Exchange. Retrieved from https://trace.tennessee.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1398&context=utk_chanhonoproj

Muhammad Ariful Islam, M. A. (2018). *A Modified and Secured RSA Public Key Cryptosystem Based on "n" Prime Numbers.* Scientific Research Publishing Inc.

Nanang Triagung Edi Hermawan, E. W. (2021). *Multi prime numbers principle to expand implementation of CRT method on RSA algorithm.* AIP Conference Proceedings 2331.

Thawte. (2013). *History of Cryptography.* Thawte, Inc.

12. Appendix

12.1. Python Code for encryption and decryption using original RSA

```
#!/usr/bin/env python3
from math import gcd

# Change the value of p and q as per your need
p, q = 11, 17
n = p * q
phi = (p-1) * (q-1)

# Select random prime number for public exponent
e = 7

# Checking if the selected prime number for public exponent qualifies the condition or not
if e > 1:
    if gcd(phi, e) == 1:
        print(f"Public Key: {e} allowed")

    else:
        print(f"Public key: {e} not allowed")
        exit()
else:
    print("Public key cannot be less than or equal to 1.")
    exit()

# Generating the private key using the public exponent and phi
for i in range(1, phi):
    if (e * i) % phi == 1:
        d = i
        print(f"Private key: {d}")
        break
```

```
# Function to encrypt the message
def enc(m):
    c = (m ** e) % n
    return c

# Function to decrypt the message
def dec(c):
    m = (c ** d) % n
    return m

if __name__ == '__main__':

    # Passing 65 as our message to encrypt and sending the cipher to decrypt
    c = enc(65)
    print(f"Cipher: {c}")

    m = dec(c)
    print(f"Plain text : {m}")
```

12.2. Python code for encryption and decryption using modified RSA

```
#!/usr/bin/env python3
# The variables have been set using the values of test 5

from math import gcd

# p, q, r, s has been set to 13, 17, 19, 23
p, q, r, s = 13, 17, 19, 23

n = p * q * r * s
phi = (p - 1) * (q - 1) * (r - 1) * (s - 1)
```

```
e1 = 13
e2 = 19

# Checking if the selected prime number for two public exponents qualifies the condition
or not
if e1 > 1 and e2 > 1:
    if (gcd(phi, e1) == 1) and (gcd(phi, e2) == 1):
        print(f"First Public Key: {e1} allowed")
        print(f"Second Public Key: {e2} allowed")

    else:
        print(f"Public keys: {e1},{e2} not allowed")
        exit()
else:
    print("Public keys cannot be less than or equal to 1.")
    exit()

# Generating the private keys using the public exponents and phi
for i in range(1, phi):
    if (e1 * i) % phi == 1:
        d1 = i
        print(f"First Private key: {d1}")

    if (e2 * i) % phi == 1:
        d2 = i
        print(f"Second Private key: {d2}")

# Function to encrypt the message
def enc(m):
    c = (((m ** e1) % n) ** e2) % n
    return c
```

```
# Function to decrypt the message
def dec(c):
    m = (((c ** d2) % n) ** d1) % n
    return m

if __name__ == '__main__':

    # Passing 88 as our message to encrypt and sending the cipher to decrypt
    c = enc(88)
    print(f"Cipher: {c}")

    m = dec(c)
    print(f"Plain text : {m}")
```