

# **Combination RSA and Diffie-Hellman Encryption Algorithms**

**Done By:**

**Salwa Talal(2019010881)**

**Lina Al-kumaim (2019011006)**

**Monia Al-saqqaf(2019011058)**

**Supervised by:**

**Dr. Jamil Rashid**

**Assistant Supervisor:**

**T. Ebtihal Al-Maqtari**

**A graduation project document submitted to the Department of Information Security in  
fulfillment of the requirement for a Bachelor's Degree in 2022 - 2023**

## **Supervisor Certification**

**We certify that the preparation of this project entitled**

**(Hybrid RSA and Diffie-Hellman)**

**prepared by Salwa Talal, Lina Al-kumaim and Monia Al-saqqaf**

**was mad under my supervision at A graduation project document submitted  
to the Department of Information Security in fulfillment of the requirement  
for a Bachelor's Degree in 2022 - 2023**

**Supervisor Name: Dr. Jamil Rashid.**

**Signature:**

**Date:**

## **Dedication**

**We dedicate this project to Allah, our creator, our  
strong pillar, and our source of inspiration He has  
been our strength throughout this journey,**

**We also dedicate this work to our supportive  
families. for Our professors & friends,**

**Our love for all of you cannot be defined.**

**God bless you all.**

## **Thanks and appreciation**

**We thank Allah Almighty first and foremost for the great grace that He has bestowed upon us, then we thank our beloved parents who do not cease to us for all their efforts from the moment of our birth to these blessed moments. For everyone who advised us, guided us, contributed, or directed us in preparing this research and connecting us to the required references and sources at any of the stages we went through, and we especially thank the distinguished supervisor Dr. Jamil Rashid & his assistant Miss. Ebtihal Al-Maqtari for excellent guidance, kind encouragement, scientific advice, helpful supervision, honest cooperation and good wishes instilled the strength in us to make this work possible. We would like to also thank our colleague Nooraldein Fetahi, Muhammed Murad and Leila Najib for helping, Supporting, and guiding us with advice, education, correction, and all the things they did with us. we are also pleased to thank the esteemed college administration: “University of Emirates International, Faculty of Engineering and Information Technology, Information Security.**

## Table of Contents

<b>Dedication.....</b>	<b>II</b>
<b>Thanks and appreciation.....</b>	<b>III</b>
<b>Summary .....</b>	<b>VII</b>
<b>Chapter One .....</b>	<b>1</b>
<b>Introduction .....</b>	<b>1</b>
<b>1.2 Define the problems:.....</b>	<b>2</b>
<b>1.3 Project objectives:.....</b>	<b>3</b>
<b>1.4 Definition of the Project: .....</b>	<b>3</b>
<b>1.6 The beneficiaries of the Project: .....</b>	<b>4</b>
<b>1.7 Project Limitations: .....</b>	<b>4</b>
<b>1.8 Hypotheses and Possibilities:.....</b>	<b>4</b>
<b>1.9 Scope of the project:.....</b>	<b>4</b>
<b>1.10 The methodology used in the project “Agile”:.....</b>	<b>5</b>
<b>1.11 Feasibility study: .....</b>	<b>6</b>
<b>1.11.1 Time feasibility: .....</b>	<b>6</b>
<b>1.11.2 Technical feasibility:.....</b>	<b>6</b>
<b>1.11.2.1 Hardware and Software Requirements: .....</b>	<b>6</b>
<b>1.11.2.2 Compatibility:.....</b>	<b>6</b>
<b>1.11.2.3 Key Management:.....</b>	<b>7</b>
<b>1.11.2.4 Security: .....</b>	<b>7</b>
<b>1.11.2.5 Performance: .....</b>	<b>7</b>
<b>1.11.3 Operational feasibility: .....</b>	<b>7</b>
<b>1.11.4 Economic feasibility: .....</b>	<b>8</b>
<b>1.11.5 Cultural feasibility: .....</b>	<b>8</b>
<b>Chapter Two.....</b>	<b>9</b>
<b>Background theory and previous.....</b>	<b>9</b>
<b>2.1 Introduction: .....</b>	<b>10</b>
<b>2.2 About encryption: .....</b>	<b>10</b>
<b>2.2.1 Encryption:.....</b>	<b>10</b>

2.2.2 Importance of encryption:.....	11
2.2.3 Types of encryption:.....	11
2.2.4 Side effects of encryption:.....	14
2.3 Past studies:.....	15
<i>Chapter Three</i> .....	18
Analysis.....	18
3.1 Introduction:.....	19
3.2 Collection techniques requirements: .....	19
3.3 User Requirements: .....	19
3.4 Functional requirements: .....	19
3.5 Non- functional requirements: .....	20
3.5.1 Usability:.....	20
3.5.2 Performance:.....	20
3.5.2.1 Speed:.....	20
3.5.2.2 Scalability: .....	20
3.5.2.3 Security: .....	20
3.5.2.4 Reliability:.....	21
3.5.2.5 Compatibility:.....	21
3.6 Support:.....	21
3.7 Adaptability:.....	21
3.8 Implementation: .....	21
3.9 Mechanism of Action:.....	22
3.10 Project scenario:.....	27
<i>Chapter Four</i> .....	28
<i>Design and Implementation</i> .....	28
4.1 Overview of the problem you are solving:.....	29
4.2 Design the hybrid algorithm: .....	29
4.3 Implementation details: .....	40
4.4 Algorithm architecture: .....	40
<i>Chapter five</i> .....	42
Results and Comparison.....	42
5.1 Security:.....	43
5.2 Performance: .....	43

5.3 Compatibility: .....	44
5.4 Easy of use: .....	44
5.5 Key management: .....	45
<i>Chapter six</i> .....	46
Conclusion and Future Work .....	46
6.1 Conclusion: .....	47
6.2 Suggestions and Recommendations: .....	47
6.3 References: .....	47

## List of Figures

Figure 1, Agile Methodology .....	5
Figure 2, Time feasibility .....	6
Figure 3, Flow chart of RSA algorithm .....	23
Figure 4, Flow chart of Diffie-Hellman algorithm .....	25

## List of Tables

Table 1, Economic feasibility .....	8
-------------------------------------	---

## Summary

What concerns people nowadays is how communicating with each other can be done in a very safe way without any corruption, and since there are always people who try to hack and steal information and sometimes they succeed in that, encryption is one of the best ways to protect information. Encryption can protect information from being stolen but encryption can also protect information from being compromised and modified by any unauthorized party, so our project aims to protect information by merge RSA and Diffie-Hellman algorithms, we merged them with AES algorithms to increase security and we have application to easily perform encryption and decryption operations with the regard of the performance and efficiency.



# *Chapter One*

## **Introduction**

## **1.1 Introduction:**

Securing data is a complicated task in data communication today that influences many areas; therefore, security is still one of the major methods of protecting information by transferring undefined and encrypted information not permitting unauthorized persons to obtain it. A full data protection solution needs to have confidentiality, integrity and authenticity. Cryptography is the process of designing procedures that enable information to be transmitted securely such that the intended recipient is the only person capable of obtaining this information. The basic cryptographic principle is described as: A text to be sent is identified as plaintext, then encrypted using an encryption algorithm, this process is known as encryption. The encrypted text produced is known as cipher text, and the decryption process turns them back into plaintext.

## **1.2 Define the problems:**

- **Organizations and companies are unaware of the importance of encryption which leads to exposing the data.**
- **One encryption algorithm is no longer enough.**
- **Most of the encryption algorithms become compromised.**
- **A lot of critical information and data in organizations and companies need to be encrypted securely.**
- **Sometimes unauthorized people and entities can access and read data.**
- **To prevent man-in-the-middle attacks from stilling the key.**
- **Lack of customer trust because of the lack of security.**

### **1.3 Project objectives:**

- **Increase awareness of the importance of encryption to reduce the exposure of data and information.**
- **Hybrids have more than one known algorithm to improve security.**
- **Find a new approach to encrypt that is not compromised.**
- **Make data and information secure as possible we can.**
- **Make data unreadable for hackers and any unauthorized entity.**
- **Increase customer trust by increasing security.**
- **Decrease man-in-the-middle attacks.**

### **1.4 Definition of the Project:**

**It is a system that integrates two encryption algorithms to take out a new encryption algorithm and enhance the terms of speed, security, reliability, integrity and confidentiality.**

### **1.5 General Goal:**

**Bring a new approach to encryption and enhance the efficiency of the security performance.**

## **1.6 The beneficiaries of the Project:**

- **Programmers.**
- **Organizations.**
- **Information security sector as a whole.**

## **1.7 Project Limitations:**

- **Lack of awareness**
- **The specialty is new**
- **Internet weakness**

## **1.8 Hypotheses and Possibilities:**

**We can solve the limitation by the following:**

- **Increase awareness about the importance of encryption**
- **Learn encryption extensively**
- **Make greater efforts in strengthening the internet**

## **1.9 Scope of the project:**

**On the user's device from anywhere.**

### 1.10 The methodology used in the project “Agile”:

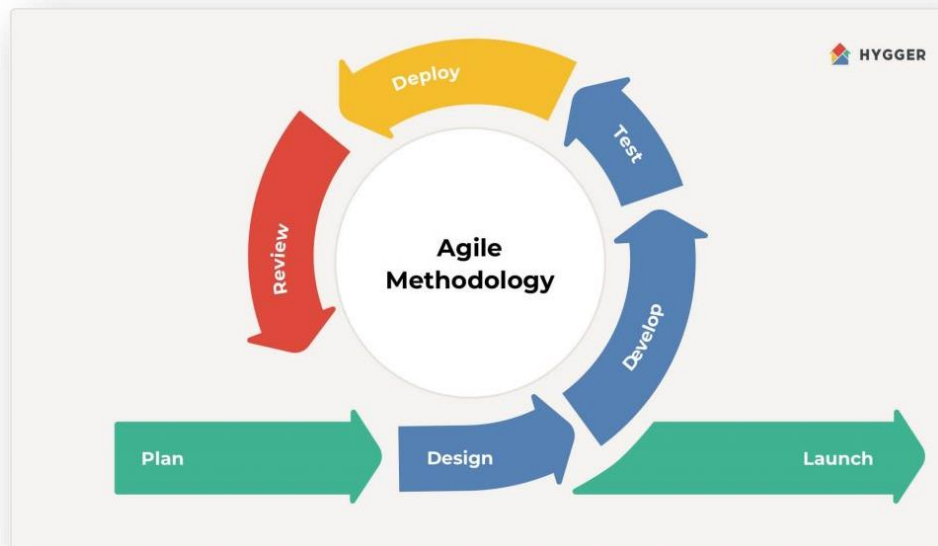


Figure 1, Agile Methodology

#### Reasons to use Agile technology:

- Complete visibility of the progress for the project in real-time.
- We can go back and modify anything to suit the project
- Change during the development process.
- Speed and flexibility.

## 1.11 Feasibility study:

### 1.11.1 Time feasibility:

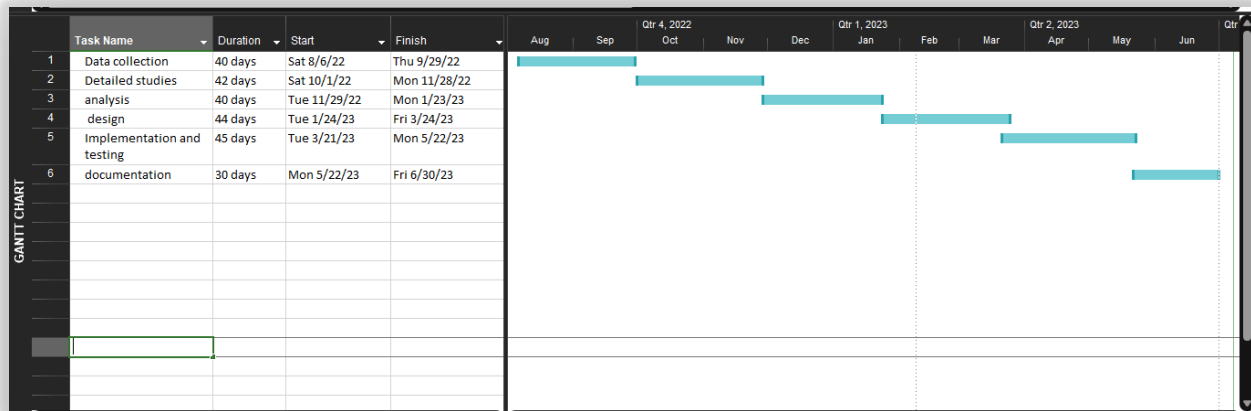


Figure 2, Time feasibility

### 1.11.2 Technical feasibility:

#### 1.11.2.1 Hardware and Software Requirements:

The hybrid encryption algorithm requires a modern computer system with sufficient processing power and memory to perform encryption and decryption operations. The algorithm is developed in Python, which is widely used and supported, making it compatible with most modern computer systems.

#### 1.11.2.2 Compatibility:

The hybrid encryption algorithm should be designed to be compatible with existing hardware and software. The algorithm should be tested to ensure compatibility with different platforms and devices.

#### **1.11.2.3 Key Management:**

The key management should be designed to securely generate and distribute keys between communicating parties. The RSA and Diffie-Hellman keys should be generated and stored securely to prevent unauthorized access.

#### **1.11.2.4 Security:**

The hybrid encryption algorithm should be designed to provide a high level of security to protect sensitive data. The encryption and decryption processes should be resistant to attacks such as brute-force attacks and cryptanalysis.

#### **1.11.2.5 Performance:**

The hybrid encryption algorithm performance has been evaluated in terms of encryption and decryption speed and found to be efficient and reasonable. 1 The system's speed and efficiency are compatible with its algorithms.

### **1.11.3 Operational feasibility:**

the operational feasibility study indicates that the hybrid encryption algorithm should be designed to meet the technical requirements of the target markets, and should be cost-effective and operationally feasible. Resource availability and operational impact should also be assessed to ensure that the system can be implemented and used effectively.

### 1.11.4 Economic feasibility:

Table 1, Economic feasibility

Physical components	Number	cost
Computer	1	700\$

### 1.11.5 Cultural feasibility:

the cultural feasibility study indicates that the hybrid encryption algorithm should be designed with cultural sensitivity in mind. The algorithm should be compatible with the cultural norms and practices of the target markets, and should comply with all legal and regulatory requirements. User acceptance, security, and privacy concerns should also be addressed in a culturally sensitive manner.



*Chapter Two*

**Background theory and  
previous**

## **2.1 Introduction:**

We can describe encryption as the complete operation of deliver text, images videos and any multimedia as it is without any modification even if they are stolen by any unauthorized party, the first known evidence of the use of encryption (in some form) was found in a carved inscription around 1900 BCE, in the main chamber of the tomb of the nobleman khnumhotep second in Egypt. The scribe used unusual hieroglyphic symbols here and there instead of the regular ones. The purpose was not to conceal the message but perhaps to alter its appearance in a way that made it seem more prestigious. Although the inscription was not a form of secret writing, it included a kind of transformation in the original text, making it the oldest known text to do so, after that cryptography continued to evolve.

## **2.2 About encryption:**

About encryption, its importance, the types of encryption, side effects and more.

### **2.2.1 Encryption:**

Encryption is used to protect data changed or compromised and works by scrambling data into a secret code that can only be unlocked with a unique digital key.

**Encrypted data can be protected while at rest on computers or in transit between them, or while being processed, regardless of whether those computers are located on-premises or are remote cloud servers.**

### **2.2.2 Importance of encryption:**

**It helps protect private information, sensitive data, and can enhance the security of communication between client apps and servers. Meaning when your data is encrypted, even if an unauthorized person or entity gains access to it, they will not be able to read it.**

### **2.2.3 Types of encryption:**

**There are two types of encryption in widespread use today: symmetric and asymmetric encryption. The name derives from whether or not the same key is used for encryption and decryption.**

**What is symmetric encryption?**

**In symmetric encryption the same key is used for encryption and decryption. It is therefore critical that a secure method is considered to transfer the key between sender and recipient.**

**What is asymmetric encryption?**

**Asymmetric encryption uses the notion of a key pair: a different key is used for the encryption and decryption process. One of the keys is typically known as the private key and the other is known as the public key.**

The private key is kept secret by the owner and the public key is either shared amongst authorized recipients or made available to the public at large.

Data encrypted with the recipient's public key can only be decrypted with the corresponding private key. Data can therefore be transferred without the risk of unauthorized or unlawful access to the data.

### **RSA ALGORITHM:**

A public key encryption algorithm developed by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1978 that became a de facto standard. RSA formed the basis for a number of encryption programs, including Pretty Good Privacy (PGP). RSA is an algorithm for public key encryption. It was the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key encryption. It involves three steps: key generation, encryption and decryption. It is still widely used in electronic commerce protocols, and its believed security depends on the difficulty of decomposition of large numbers.

### **Advantages of RSA Algorithm:**

It uses Public Key encryption which means that the text will be encrypted with someone's Public Key (which everyone knows about) but only the person intended for can read it, by using their private key (which only they know about).

- Use of public key in RSA provides digital signatures that cannot be repudiated.
- Ciphering & deciphering algorithm are same.

### **Problems in RSA Algorithm:**

**If any one of  $p$ ,  $q$ ,  $e$ ,  $d$  is known, then the other values can be calculated. So secrecy is important.**

- It is important to make sure that message length should be less than bit length otherwise the algorithm will fail.**
- Due to the usage of public key RSA is much slower than any other symmetric cryptosystems.**
- The length of plain text that can be encrypted is limited to the size of  $n=p*q$ .**

### **Diffie-Hellman algorithm:**

**Whitfield Diffie and Martin Hellman discovered what is now known as the Diffie-Hellman (DH) algorithm in 1976. It is an amazing and ubiquitous algorithm found in many secure Connectivity protocols on the Internet. In an era when the lifetime of “old” technology can sometimes be measured in months, this algorithm is now celebrating its 25th anniversary while it is still playing an active role in important Internet protocols. DH is a method for securely exchanging a shared secret between two parties A and B over a public network and each holding public/private key to agree on a shared secret value. This shared secret is important between two parties who may not have ever communicated previously, so that they can encrypt their communications. As such, it is used by several protocols, including Secure Sockets Layer (SSL), Secure Shell (SSH), and Internet Protocol Security (IPSec). These protocols will be discussed in terms of the technical use of the DH algorithm and the status of the protocol standards established or still being defined. Diffie–Hellman establishes a shared secret key that can be used for secret communications by exchanging data over a public network.**

### **Advantages of Diffie-Hellman Algorithm:**

- **No known successful attack strategies until now, so it is secure.**
- **Diffie-Hellman protocol generates a “shared\_secret” an identical cryptographic key shared by each side of the communication.**

### **Problems in Diffie-Hellman Algorithm:**

- **It is easily susceptible to man-in-the-middle attacks.**
- **The algorithm cannot be used to encrypt messages.**
- **There is also a lack of authentication.**
- **The computational nature of the algorithm could be used in a denial-of-service attack very easily.**

### **2.2.4 Side effects of encryption:**

#### **Pros:**

- 2- Limit the Reach of the Breach**
- 2- you do not have to be a cryptographer**

#### **Cons:**

- 2- Lost Key = Lost Sanity**
- 2- Human Passwords Are Easy to Crack**

## **2.3 Past studies:**

### **2.3.1 Enhanced Multistage RSA Encryption Mode**

**In 2020 Mays M. Hoobi, Sumaya S. Sulaiman, Inas Ali AbdulMunem have developed an approach that improve the security of the RSA algorithm. By increasing the complexity and search space of RSA algorithm against brute force attack in addition to security enhancement was satisfied by applying four cases with using different cryptography algorithms. By applying four cases using different cryptography algorithms. This four cases included case1: enhanced the security of RSA by using Optimal Asymmetric Encryption Padding (OAEP), case2: combining of the two most important algorithms RSA and Diffie-Hellman (D-H), case3: for increasing complexity and obtaining high level of security the two above cases (case1& case2) were concatenated, finally for most complexity and obtained highest security level with increasing search space of RSA case4 was applied. Case4: contained implementation of case3 in addition to apply new level of security by adding another cryptography algorithm called HiSea algorithm. The results of using multiple cryptography algorithms in each case of the above four cases respectively improved the security level by increasing the complexity and key search space that lead to protecting the security goals against the attackers.**

**Advantages of this research:**

- It supports big data block sizes.
- It offers strict protection against attacks of difference.
- Large number of possibilities need to be tested to dedicate the right key.
- The multi-layer encryption is considered as protected encryption algorithm.
- It is computationally safe against attacks by brute force as it takes a lot of time to test all possible keys.
- Improving the robustness of the algorithm by using a set of different cryptography algorithms instead of one algorithm.

**2.3.2 Hybrid encryption algorithm:**

In 2014 Gaurav R. Patel, Prof. Krunal Panchal tries to improve an approach RSA and DH algorithms by adding one more operation which is bitwise XOR operation. This operation is performed after the message is converted into cipher text. In this proposed approach first, we choose two prime numbers and find out the Encryption and decryption key exponents which will be used for the encryption and decryption process. For Diffie Hellman algorithm we select A and B. R is a random prime number generated by the system automatically. The public number is generated by the Diffie Hellman algorithm. By using this public number, we can generate secret key  $K_A$  and  $K_B$ . This will be used to perform XOR operation. At the sender side the encryption is done using encryption algorithm. When the encryption process



completes XOR operation perform between cipher text and the first secret key. After that operation, the secret message is sent over the medium. At the receiver side the XOR operation is again performed between the second secret key and the secret message which is sent by the sender. Using this operation, we get the original cipher text. We can decrypt the cipher text using a decryption algorithm and get the original message sent by the sender.

**Advantages of this research:**

- can improve security of message
- complexity of the message is also increased
- provides more security compared to normal RSA algorithm

**2.3.3 Enhancement of Security of Diffie-Hellman Key Exchange Protocol using RSA Cryptography:**

**Chiradeep Gupta, N V Subba Reddy have improved an approach**

**This paper proposes a model of integrating the public key RSA cryptography system with the DH key exchange to prevent the MITM attack. The performance of the proposed work has been compared to the DH Key Exchange algorithm as well as RSA Cryptosystem to conclude the effectiveness of the proposed model.**

# *Chapter Three*

## **Analysis**

### **3.1 Introduction:**

**In this chapter we will address and explain the analysis phase, where users will be clarified, the functional and non-functional requirements will also be clarified and some graphics will help us to explain.**

### **3.2 Collection techniques requirements:**

**Researches: we have lauded some websites and articles to search about the mechanisms of RSA and Deffie-hellman algorithms, how possibly we can integrate them and the flow charts.**

### **3.3 User Requirements:**

**they must have a good previous knowledge of encryption methods and programming.**

### **3.4 Functional requirements:**

- Key generation: Both RSA and Diffie Hellman require the generation of public and private keys for encryption and decryption.**
- Key exchange: Diffie Hellman is used to securely exchange keys between two parties without the need for pre-shared keys.**

- **Encryption/decryption:** RSA is used for encryption and decryption of messages using the exchanged keys.
- **Authentication:** RSA can be used for digital signatures to authenticate the sender of a message.

### **3.5 Non- functional requirements:**

#### **3.5.1 Usability:**

The algorithm must be easy to use for all programmers and anyone that has previous knowledge of computers, so our algorithm depends on easy to use, so it offers a drawing interface and easy-to-use symbols that do not form a difficulty for users.

#### **3.5.2 Performance:**

##### **3.5.2.1 Speed:**

The hybrid algorithm is fast enough to provide real-time encryption and decryption.

##### **3.5.2.2 Scalability:**

The algorithm can handle large amounts of data without compromising its performance.

##### **3.5.2.3 Security:**

The algorithm is secure enough to prevent unauthorized access and protect sensitive information.

#### **3.5.2.4 Reliability:**

**The algorithm is reliable and consistent in its performance, even under heavy loads or adverse conditions.**

#### **3.5.2.5 Compatibility:**

**The algorithm is compatible with different platforms and systems, allowing for seamless integration and interoperability.**

### **3.6 Support:**

**The algorithm supports the Windows operating system in all its versions and supports Linux systems.**

### **3.7 Adaptability:**

**The algorithm can be added to any technology as per business requirements.**

### **3.8 Implementation:**

**Will use Python to implement this algorithm.**

### **3.9 Mechanism of Action:**

#### **RSA ALGORITHM:**

##### **Steps of Algorithm for Key Generation:**

- 1. Choose two distinct prime numbers P and Q.**
- 2. Calculate  $N = P \times Q$ . (n is used as mod for both the public and private keys)**
- 3. Select the public key (i.e. encryption key) E such that it is not a factor of  $(P - 1)$  and  $(Q - 1)$ .**
- 4. Select the private key (i.e. the decryption key) D such that the following equation is true  $(D \times E) \bmod (P - 1) \times (Q - 1) = 1$ .**
- 5. For encryption, calculate the cipher text CT from the plain text PT as follows:  $CT = P^E \bmod N$ .**
- 6. Then send CT as the cipher text to the receiver.**
- 7. For decryption, calculate the plain text PT from the cipher text CT as follows:  $PT = CT^D \bmod N$ .**

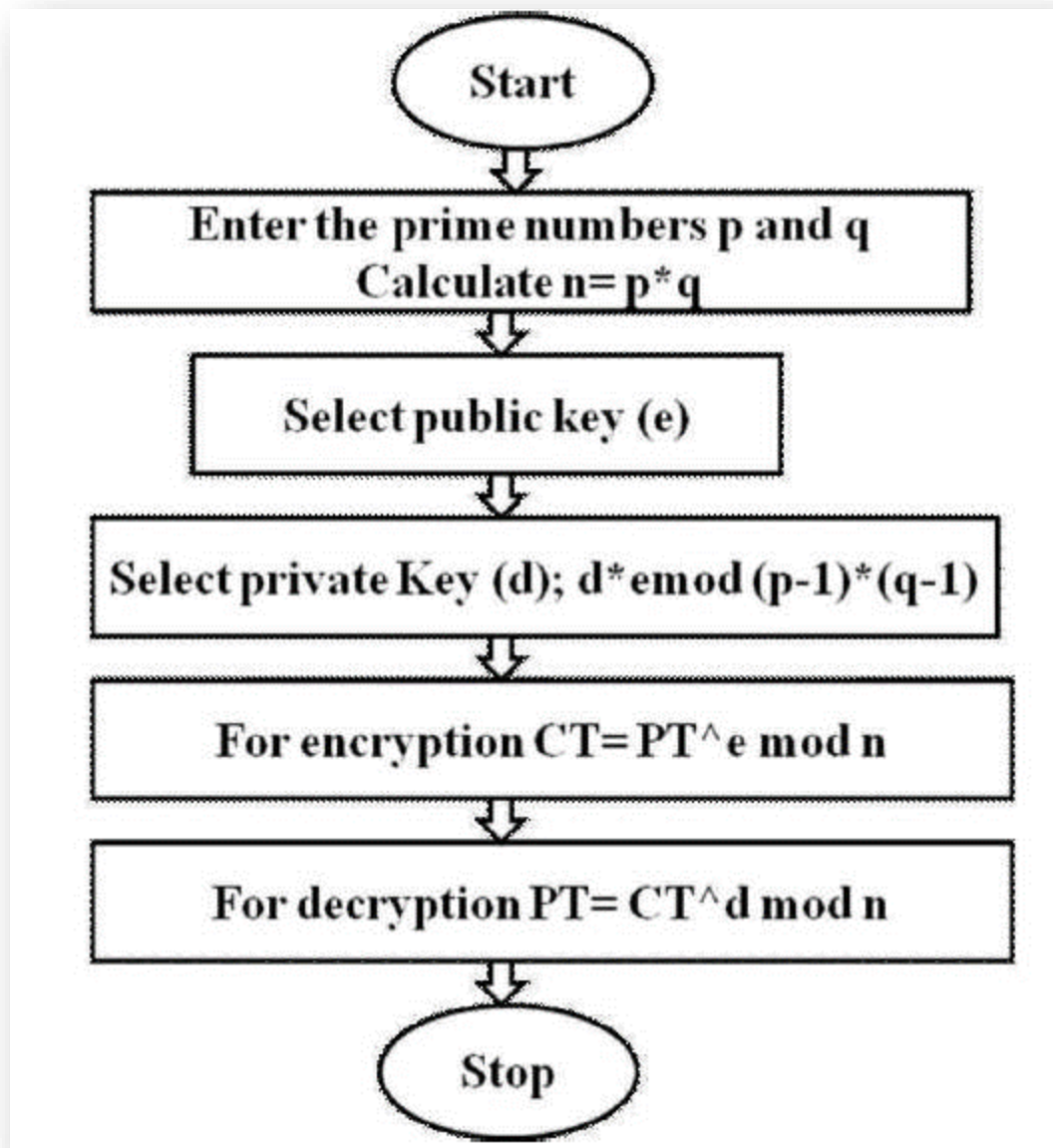


Figure 3, Flow chart of RSA algorithm

## **DIFFIE–HELLMAN ALGORITHM:**

**Steps of this algorithm are as:**

- 1. Taking two numbers “P” and “G” “P” is a large prime number “G” is called the base.**
- 2. Picks a secret number “A” as first secret number = A, then picks another secret number “B” as second secret number = B.**
- 3. Computes first public number  $X = GA \bmod P$ , and public number = X.  
Then computes second public number  $Y = GB \bmod P$ , and public number = Y.**
- 4. Exchange their public numbers.**
- 5. First knows P, G, A, X, Y, second knows P, G, B, X, Y.**
- 6. Computes First session key as  $KA = YA \bmod P$  OR  $KA = (GB \bmod P) A \bmod P$  OR  $KA = (GB) A \bmod P$  OR  $KA = GBA \bmod P$ .**
- 7. Computes second session key as  $KB = XB \bmod P$  OR  $KB = (GA \bmod P) B \bmod P$  OR  $KB = (GA) B \bmod P$  OR  $KB = GAB \bmod P$ .**
- 8. Fortunately for Both by the laws of algebra, first session key “KA” is the same as Second session key “KB”, or  $KA = KB = K$ .**
- 9. Know we have both the secret value as “K”.**



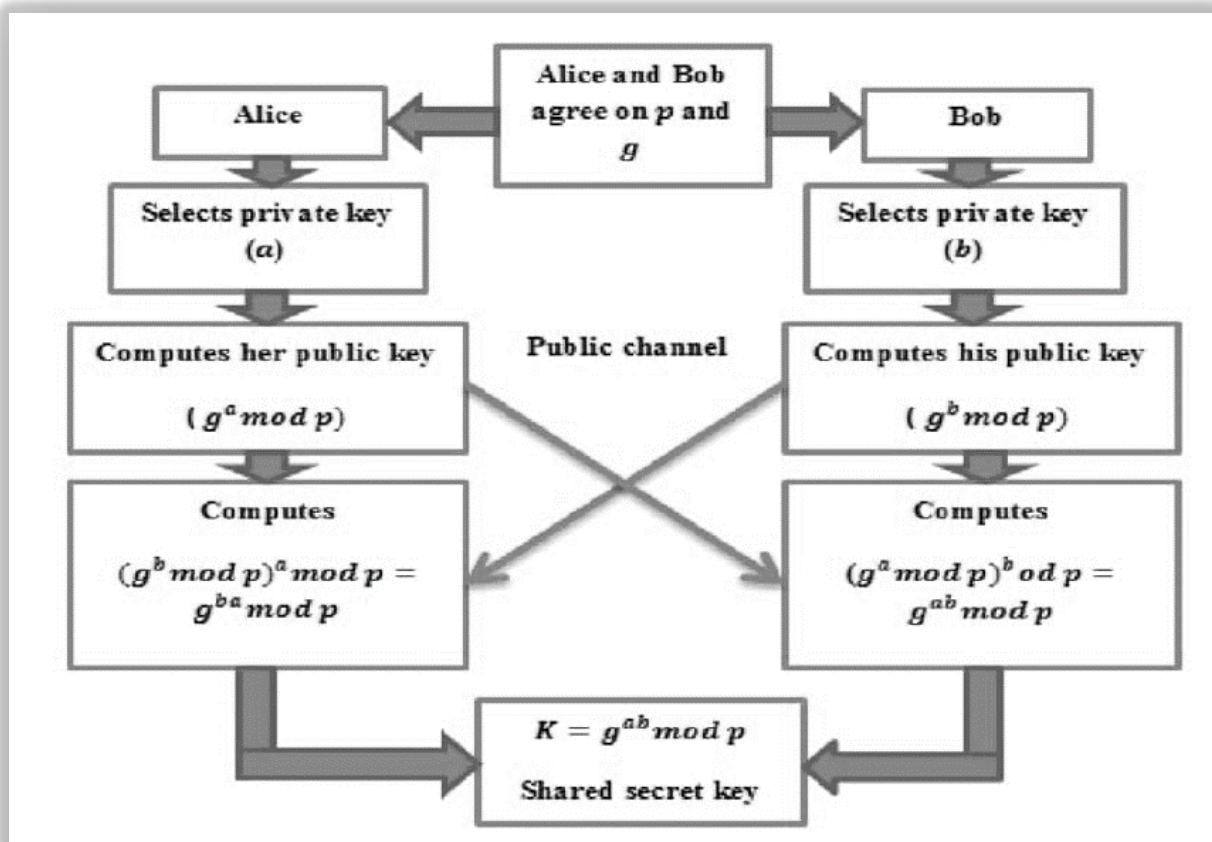
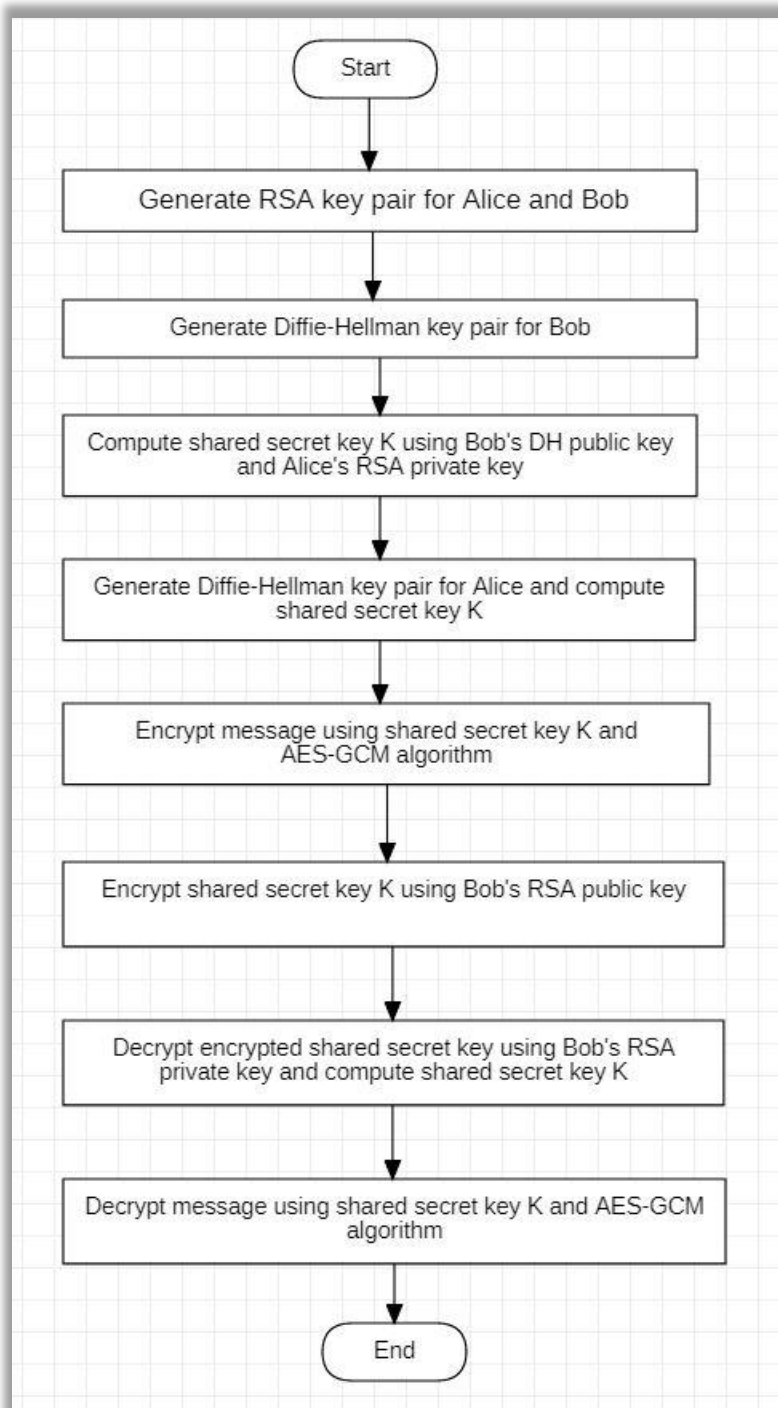


Figure 4, Flow chart of Diffie-Hellman algorithm

## Hybrid RSA and Diffie-Hellman



### 3.10 Project scenario:

The project is a hybrid encryption algorithm using RSA and Diffie-Hellman is a secure and efficient way for two parties. The algorithm uses both symmetric and asymmetric encryption to combine their strengths. Alice and Bob first generate RSA key pairs, consisting of a public key and a private key. Bob also generates a Diffie-Hellman key pair, consisting of a private key and a public key. Alice generates her own Diffie-Hellman key pair. Bob sends his Diffie-Hellman public key to Alice, and Alice uses Bob's public key and her private key to compute a shared secret key  $K$  using the Diffie-Hellman key exchange algorithm. Alice encrypts the message using the symmetric AES-GCM algorithm and the shared secret key  $K$ . Alice encrypts the shared secret key  $K$  using Bob's RSA public key. Alice sends the encrypted message and the encrypted shared secret key to Bob. Bob uses his private key to decrypt the encrypted shared secret key and computes the shared secret key  $K$  using his private key, Alice's public key, and the Diffie-Hellman key exchange algorithm. Bob decrypts the message using the shared secret key  $K$  and the AES-GCM algorithm and reads the decrypted message. The algorithm is implemented using various libraries, including ``random``, ``math``, ``Cryptodome``, and ``os``, and a GUI is created using the ``Tkinter`` module.

# *Chapter Four*

## *Design and Implementation*

## **4.1 Overview of the problem you are solving:**

**In today's digital age, secure communication is a critical need. With the rise of electronic communication, the exchange of sensitive information has become more commonplace. Encryption algorithms are used to protect sensitive data from unauthorized access. However, existing encryption algorithms have limitations, such as vulnerability to attacks, slow processing, and high computational requirements. These limitations make it difficult to implement encryption on a large scale, and they can also compromise the security of the encrypted data.**

**The need for a more secure and efficient encryption algorithm has led to the development of hybrid encryption algorithms. Hybrid encryption algorithms combine the strengths of multiple encryption algorithms to provide enhanced security and better performance.**

**In this project, we aim to develop a hybrid encryption algorithm that combines RSA and Diffie Hellman to provide enhanced security and better performance.**

## **4.2 Design the hybrid algorithm:**

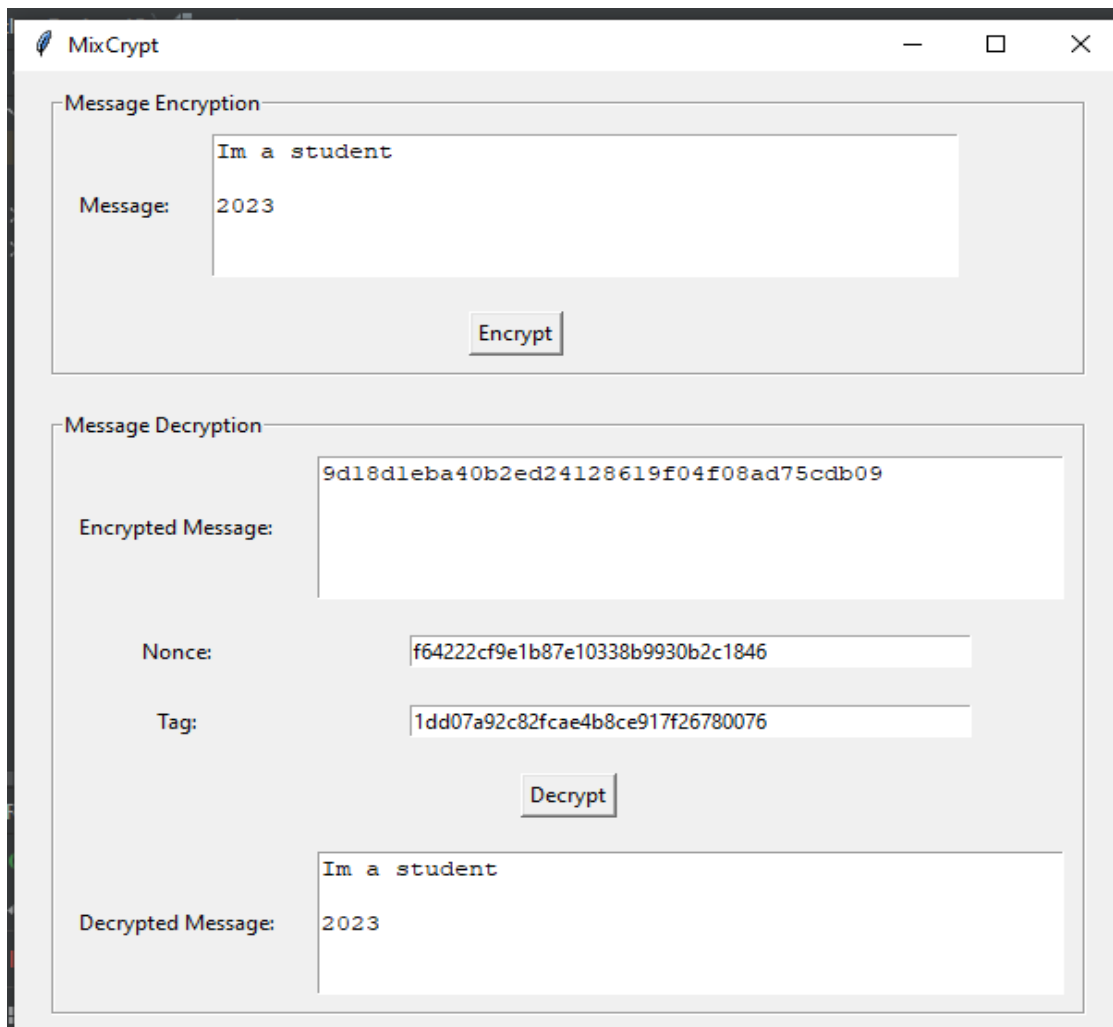
**Our hybrid encryption algorithm combines RSA and Diffie Hellman.**

**In our algorithm, Alice and Bob generate their RSA key pairs and generate Diffie Hellman key pairs. then compute shared secret key  $K$  using Bob's DH public key and Alice's RSA private key. Alice encrypts the shared secret key using Bob's RSA public key and sends it to Bob. Bob decrypts the shared secret key using his RSA private key and computes the shared secret key using his**

Diffie Hellman key pair and Alice's RSA public key. Once the shared secret key is established, Alice and Bob can use it to encrypt and decrypt messages using AES-GCM.

The reason we chose RSA and Diffie Hellman is that RSA is an asymmetric key encryption algorithm that provides secure key exchange and digital signatures. Diffie Hellman is algorithm that provides secure key exchange over an insecure channel. By combining these two algorithms, we can take advantage of their complementary strengths and enhance the security of our hybrid algorithm.

#### 4.2.1 Design in python



The screenshot displays the MixCrypt application window, which is divided into two main sections: Message Encryption and Message Decryption.

**Message Encryption Section:**

- Message:** A text input field containing "Im a student".
- Message:** A text input field containing "2023".
- Encrypt:** A button to perform the encryption operation.

**Message Decryption Section:**

- Encrypted Message:** A text input field containing the hexadecimal string "9d18d1eba40b2ed24128619f04f08ad75cdb09".
- Nonce:** A text input field containing the hexadecimal string "f64222cf9e1b87e10338b9930b2c1846".
- Tag:** A text input field containing the hexadecimal string "1dd07a92c82fcae4b8ce917f26780076".
- Decrypt:** A button to perform the decryption operation.
- Decrypted Message:** A text input field containing the original message "Im a student" and the year "2023".

### 4.2.2 Design in Web

Interface has two buttons encrypt and decrypt.

تشفير

الرسالة الأصلية :

تشفير

الرسالة المشفرة:

تأكيد

The encrypt interface has two text area the first one to enter the text that user want to encrypt. it the second text area display the text after being decrypted.

## تشفير

الرسالة الأصلية :

Hi dear  
I hope you are doing great, I am writing this letter to tell you that I  
.am moving next month from Roma to Paris

تشفير

الرسالة المشفرة:

d89f452d99f93b4ef068573d3aeaa2c0e7de97c838a1d5c14c4671cf  
871e740f326ae602ce7430a7b22074b30c594d0fa57778170aef49fb  
48c82d5207022b5cc02315083dd8c6d3b55e365c6ee30439b7ed678  
be90c8de71c92415b27154135f9b151352583f61d406a0fce4f64e92  
bc9b547143f6ef68dcf55

تأكيد



The password interface adds another level of security so that no one could decrypt the ciphertext unless he or she had the password that they agreed in advance.

الرسالة الأصلية :

ادخل كلمة السر للتأكيد

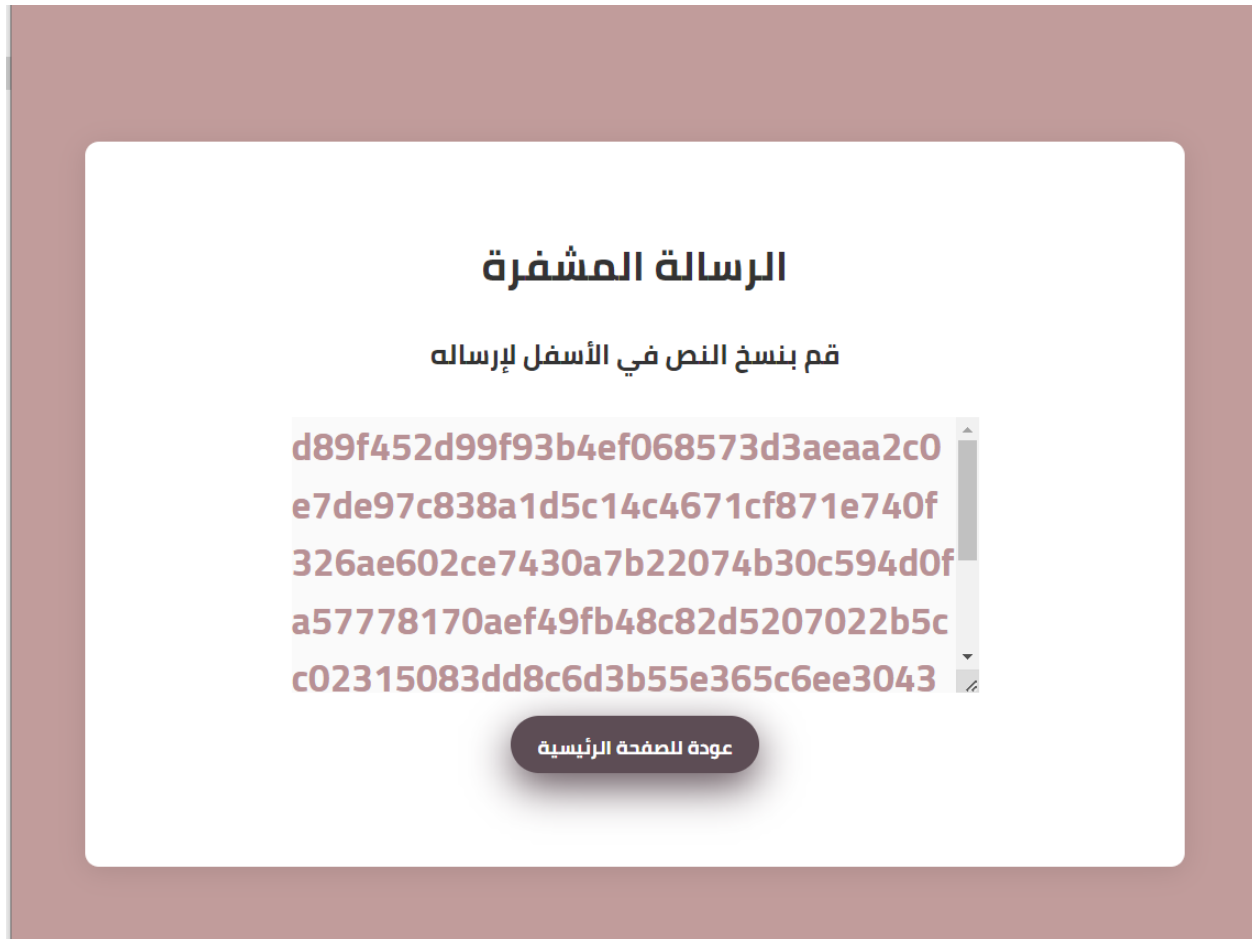
تأكيد الغاء

الرسالة المشفرة:

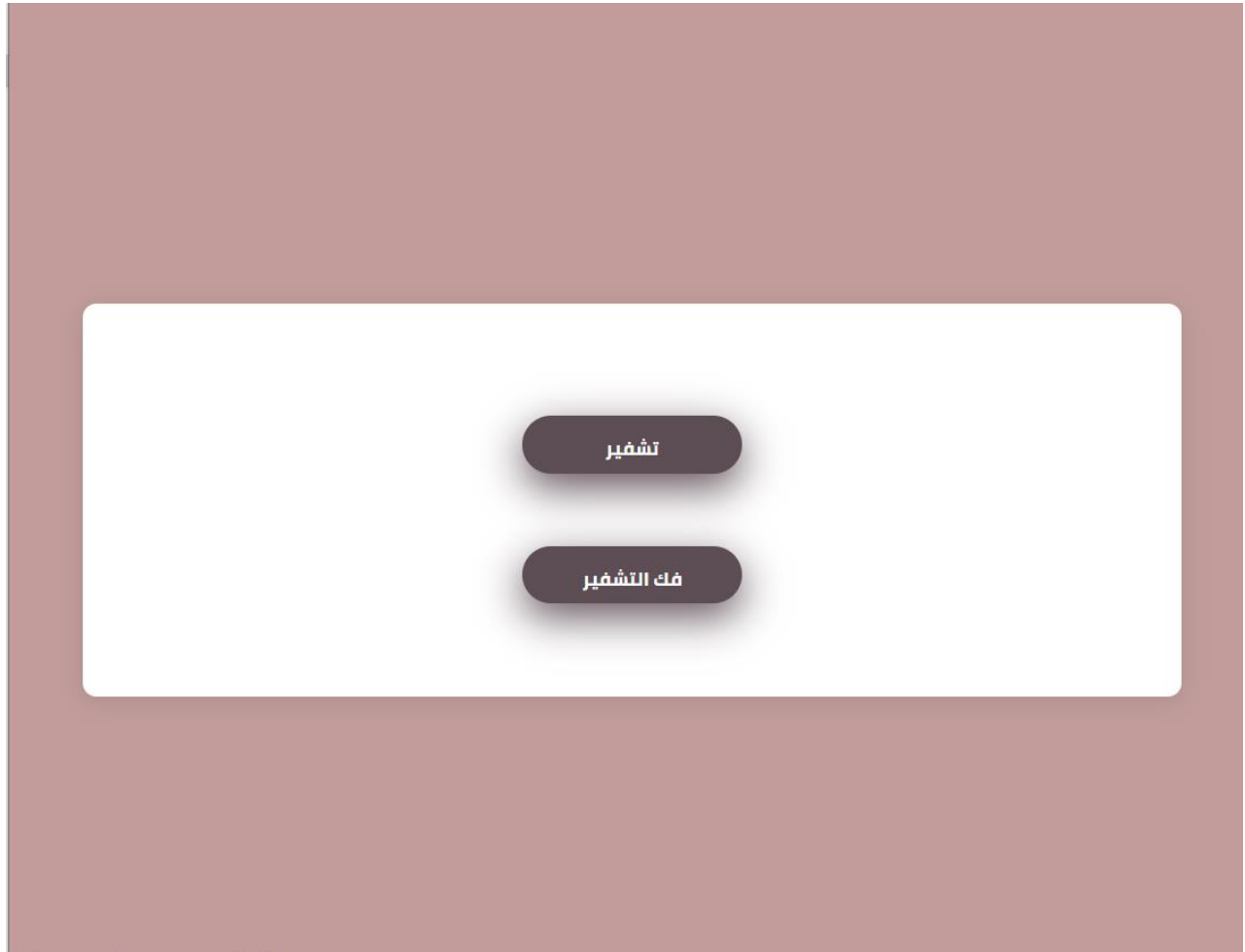
7dd4a0170cd9e6b450f9fdf2e53df53997624c79fbb423845795325d  
630074339a14ec490b09e667eac101cda85a76413e632553510584  
be45f689ed9dfa7d70e9dba8cf1b14614cab3682c759b2d6417c7d52  
82c87c61672d207a38e4130a0df1b28a2d5886082a136b55c637eba  
61798aa351a03b809a381

تأكيد

The text after being encrypted



Interface has two buttons encrypt and decrypt.



**the decryption interface requires the password that two parties agreed in advance**



The image shows a decryption interface. It has a title 'فك تشفير' (Decrypt) in Arabic. Below the title is a password input field with a dotted line indicating the password length. Below the input field is a button labeled 'دخول' (Enter) in Arabic. The interface is set against a dark red background.

The decryption interface has two text areas first one is to past the ciphertext the second one is the original plaintext.

## فك تشفير

الرسالة المشفرة:

81c5b644fb1daa27a18f7d5066228d5d8fac07fed1910a55d96a72a6  
34d328bf62c7676cc20ee5cc3f6db51bb6a26629a7398a42e3b9b81  
2c4e7417e3bf710fcb32d9262c1b2042d3a1b2b4a49e5dae537a1bec  
f4abbe32128cbf93eca970601e826ecc3a829427df4db05353cedd6c  
2c62e3cefd493ba8df7

فك التشفير

الرسالة الاصلية:

Hi dear  
I hope you are doing great, I am writing this letter to tell you that I  
.am moving next month from Roma to Paris

تم

### 4.2.3 Design in Android

The encrypt interface has two text area the first one to enter the text that user want to encrypt it the second text area display the text after being decrypted.



Interface has two buttons encrypt and decrypt.



### **4.3 Implementation details:**

We implemented our algorithm in Python using the Cryptodome library. Our code generates RSA and Diffie Hellman key pairs, encrypts and decrypts messages using AES-GCM, and provides a graphical interface for users to communicate securely. We used the tkinter library to develop the GUI. Our code also includes error handling and validation to ensure secure communication.

The reason we chose Python is that it is a popular programming language for cryptography and has extensive libraries for encryption and decryption. We chose the Cryptodome library for its ease of use and compatibility with Python 3. We also chose tkinter for the GUI because it is a lightweight and easy-to-use graphical interface library that is included with Python.

### **4.4 Algorithm architecture:**

Our algorithm consists of three main components: the key generation component, the encryption and decryption component, and the GUI component. The key generation component generates the RSA and Diffie Hellman key pairs. The encryption and decryption component encrypts and decrypts messages using AES-GCM and implements our hybrid algorithm. The GUI component provides a user-friendly interface for users to input messages, encrypt and decrypt them, and view the results.

The reason we divided our algorithm into three main components is to make it modular and easy to maintain. By separating the key generation, encryption and decryption, and GUI components, we can make changes to one component



without affecting the others. This modularity also makes our algorithm more scalable and flexible.

To evaluate the performance and security of our algorithm, we compared it with existing encryption algorithms, such as RSA, Diffie Hellman, and AES. We measured the encryption and decryption times for different message sizes and found that our algorithm was faster than RSA and Diffie Hellman alone and more secure than AES alone. However, we also found that our algorithm has some weaknesses, such as vulnerability to replay attacks, and we suggest further research in this area.

The reason we conducted an evaluation is to determine the effectiveness of our algorithm and identify areas for improvement. By comparing our algorithm with existing algorithms, we can assess its strengths and weaknesses and determine its potential impact on secure communication. We also suggest further research in areas such as replay attacks to improve the security of our algorithm.

# *Chapter five*

## **Results and Comparison**

## **5.1 Security:**

- RSA:** RSA is considered a secure public-key cryptography algorithm when used with sufficiently long key sizes. However, RSA is vulnerable to attacks if the key size is too small.
- Diffie-Hellman:** Diffie-Hellman is considered a secure key exchange algorithm when used with sufficiently large prime numbers and secure random number generation. However, Diffie-Hellman is vulnerable to man-in-the-middle attacks if not used correctly or if the shared secret key is compromised.
- AES:** AES is considered a secure symmetric-key encryption algorithm when used with sufficiently long keys and secure random number generation. However, AES is vulnerable to attacks if the key is too small.
- Hybrid Encryption Algorithm:** Our hybrid encryption algorithm provides a high level of security by combining RSA and Diffie-Hellman for key exchange and AES for message encryption. The use of RSA and Diffie-Hellman provides strong authenticity, integrity, and confidentiality, while the use of AES provides strong security and efficiency.

## **5.2 Performance:**

- RSA:** RSA can be slow when encrypting large amounts of data, particularly when used with long key sizes.
- Diffie-Hellman:** Diffie-Hellman is relatively fast and efficient for key exchange, particularly when used with large prime numbers.

- **AES:** AES is fast and efficient for encrypting and decrypting large amounts of data.
- **Hybrid Encryption Algorithm:** Our hybrid encryption algorithm provides a good balance of security and performance, particularly when encrypting and decrypting large amounts of data using AES. However, the key exchange process involving RSA and Diffie-Hellman may be slower compared to other key exchange algorithms.

### **5.3 Compatibility:**

- **RSA:** RSA is widely supported by cryptographic libraries and software, and can be used on a variety of platforms and devices.
- **Diffie-Hellman:** Diffie-Hellman is widely supported by cryptographic libraries and software, and can be used on a variety of platforms and devices.
- **AES:** AES is widely supported by cryptographic libraries and software, and can be used on a variety of platforms and devices.
- **Hybrid Encryption Algorithm:** Our hybrid encryption algorithm may require specific cryptographic libraries or software to be installed on the system, which may limit its compatibility with certain platforms and devices.

### **5.4 Easy of use:**

- **RSA:** RSA can be challenging to use for non-expert users, particularly when generating and managing keys.
- **Diffie-Hellman:** Diffie-Hellman can be challenging to use for non-expert users, particularly when selecting prime numbers and generating keys.
- **AES:** AES can be relatively easy to use for non-expert users, particularly when using standardized modes of operation.
- **Hybrid Encryption Algorithm:** Our hybrid encryption algorithm may be challenging to use for non-expert users, particularly when generating and managing keys, and configuring the key exchange and encryption process.

### **5.5 Key management:**

- **RSA:** RSA key management involves generating and securely storing public and private keys, and revoking or updating keys when necessary.
- **Diffie-Hellman:** Diffie-Hellman key management involves securely generating and exchanging shared secret keys, and revoking or updating keys when necessary.
- **AES:** AES key management involves securely generating and distributing symmetric keys, and revoking or updating keys when necessary.
- **Hybrid Encryption Algorithm:** Our hybrid encryption algorithm involves key management for both RSA/Diffie-Hellman and AES, which may require additional steps for generating, exchanging, and securely storing keys.

# *Chapter six*

## **Conclusion and Future Work**

## 6.1 Conclusion:

**Our project merge three different encryption algorithms RSA, Diffie-Hellman and AES to increase the level of security, reliability and privacy so any user could use it with high level of confidence and with our application and website this job will be done so easily**

## 6.2 Suggestions and Recommendations:

- **We suggest that encryption should be added to files and folders.**
- **We suggest to use this encryption algorithm for airports systems.**
- **We suggest to marge another encryption algorithm to our algorithm with the regard of the speed.**

## 6.3 References:

- Design and Implementation of Rivest Shamir Adleman's (RSA) Cryptography Algorithm in Text, File Data Security.
- GENERALIZED RSA CIPHER AND DIFFIE-HELLMAN, PROTOCOL, LUKASZ MATYSIAK.
- Hybrid Encryption Algorithm 1Gaurav R. Patel, 2Prof. Krunal Panchal, 1PG Scholar, 2Assistant professor LJIET, Ahmedabad.