ADVANCED-DATA SECURITY USING MULTIPLE CRYPTOGRAPHIC ALGORITHMS OVER MULTIPLE LAYERS AND SECURED KEY SHARING

A PROJECT REPORT

Submitted by

TANMOY SEN GUPTA [Reg No: RA1711003030606]
RITIK SRIVASTAVA [Reg No: RA1711003030593]

Under the guidance of

Dr. K. MARIMUTHU, Ph.D
(Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2021

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**ADVANCED-DATA SECURITY USING MULTIPLE CRYPTOGRAPHIC ALGORITHMS OVER MULTIPLE LAYERS AND SECURED KEY SHARING**" is the bonafide work of " **TANMOY SEN GUPTA [Reg No: RA1711003030606], RITIK SRIVASTAVA [Reg No: RA1711003030593], , ,** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. K. MARIMUTHU, Ph.D **GUIDE** Professor Dept. of Computer Science & Engineering	Dr. R.P.Mahapatra **HEAD OF THE DEPARTMENT** Dept. of Computer Science & Engineering
Signature of the Internal Examiner	Signature of the External Examiner

ABSTRACT

Security of Data is one of the most important aspects of one's digital presence. Many Encryption Algorithms are used in various apps and services to secure data, like AES, DES, Blowfish, RSA, and Triple-DES among others. With the advent of new and sophisticated technologies, these existing data security systems are becoming less efficient in protecting data. Advanced data security protocols used in various applications like WhatsApp and Telegram are becoming inadequate to protect the privacy of an individual. With advancements in Hardware, the time required to break a cryptographic system is becoming even lesser than earlier.

Thus the requirement of a multi-layered encryption protocol that utilizes multiple advanced cryptographic algorithms implemented in a cascading manner is ever more necessary. In this project, data will be encrypted using multiple algorithms like AES, RSA, and Blowfish. The keys that will be used for encryption will also be secured similarly and transmitted only after authentication. Multi-layered encryption protocols are not widely used in current systems, hence this project will address the problems faced by standalone systems used for data security.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, Dr.K. Marimuthu his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

Author

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

φ	Euler’s Totient Function
$\oplus$	Append to List

CHAPTER 1

INTRODUCTION

1.1 Cryptography

Cryptography is the study of techniques and methods of securing data and communication between different parties, from malicious intruders, eavesdroppers, and adversaries. Cryptography involves various aspects of securing data communication like creating protocols to facilitate secured and safe connectivity, analyzing cryptosystems for vulnerabilities, and performance and checking for the fulfillment of information security aspects like integrity, authenticity, confidentiality, and non-repudiation. Cryptography is the backbone of Information security and Information Security is essential for a user's digital presence.

Cryptography is primarily associated with Encryption. Encryption is the procedure of converting readable information/data to unreadable gibberish. Encryption is facilitated by the use of Cryptographic Algorithms or Cryptosystems. Modern cryptosystems are heavily dependent on mathematical theories and functions as well as theoretical computer science practices. Encryption is facilitated using mathematical functions that require a user-known secret, called the Key. Decryption reverses what encryption does, that is, decryption converts the unintelligible gibberish back into readable information. Using the Key.

Cryptography algorithms can be classified into two categories - Asymmetric Cryptography and Symmetric-Key Cryptography. Symmetric-Key algorithms for Cryptography uses the identical keys for both encryption, as well as for decryption. AES, Blowfish are some examples of symmetric cryptography. Whereas Asymmetric Cryptography, also known as Public-Key encryption, uses different keys - Public-Key and Private-Key for encryption, and decryption, respectively.

1.2 History of Cryptography

A brief history of cryptography (till 2001) is given in table below.

Table 1.1: History of Cryptography

100 A.D	Ceasar Cipher
1553	Inventioon of Vigenère cipher
1835	Morse code developed by Samuel Morse
1854	Invention of Playfair Cipher by Charles Wheatstone
1883	Auguste Kerckhoffs publishes his book on Laws of Cryptography, *La Cryptographie militare*
1932	Enigma Machine Broken
11948	Claude Shannon publishes his Paper on Information Theory
1974	Block cipher, Feistel-network designed by Horst Feistel
1976	• DES-Data Encryption Standard, a Symmetric-Key Cipher based on Fiestel Networks, published as an Information Processing standard by the NBS in the United States. • Diffie–Hellman key exchange Published
1977	Asymmetric-Key Encryption Algorithm, RSA invented by Adi Shamir, Ron Rivest, and Leonard Adleman
1994	Netscape releases SSL or Secure Sockets Layer encryption protocol.
1995	SHA (Secured Hashing Algorithm) is published
2001	Rijndael replaces DES to become the new Information Security Standard, the Advanced Encryption Standard, by the NIST.

1.3 Blowfish Algorithm

Blowfish, a Symmetric-Key Block Cipher, was developed by B. Schneier in the year 1993. Blowfish algorithm has 64 Bits block size and variable key length of 32 to 448 Bits. It is particularly known for its features like complicated key schedules and key-dependent s-boxes. Being a Feistel cipher it has 16 rounds. Each round, consists of four steps. In nth round, the left half of the block and the nth element in the subkey-array are XORed followed by passing it to the round function F. The return from the function F and the right half of the initial block are XORed and then swapped. The round function F divides the 32-bit input into four 8-bit blocks that are then fed to 4 different S-Boxes. The returns from the 1^{st} and 2^{nd} s-box are added and the return is XORed with the returns from the 3^{rd} s-box and again added with the output of the 4^{th} s-box. It is one of the fastest algorithms for encryption.

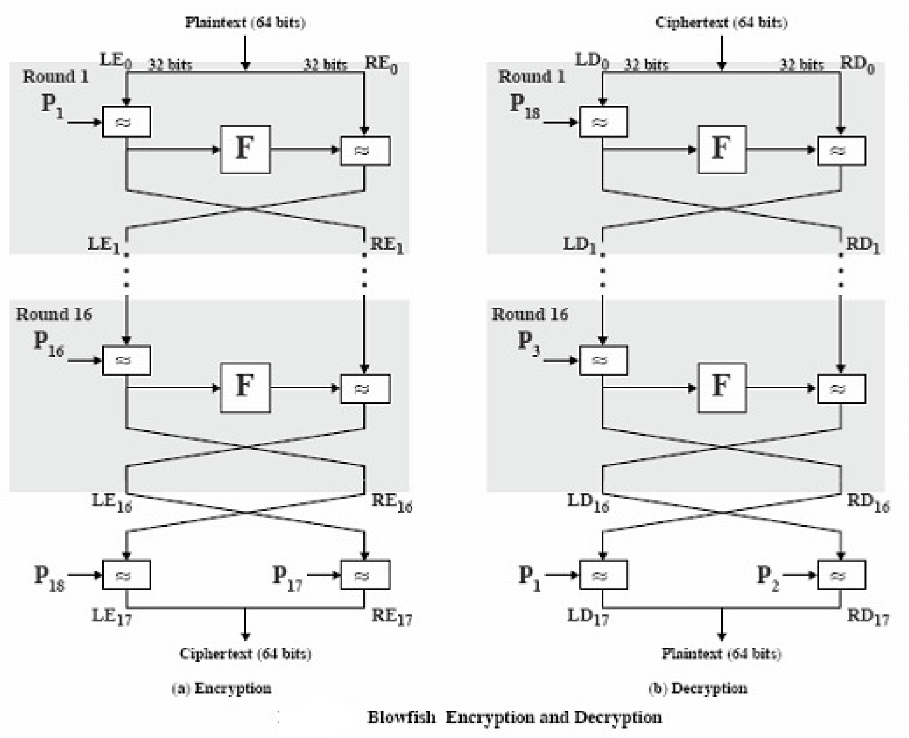
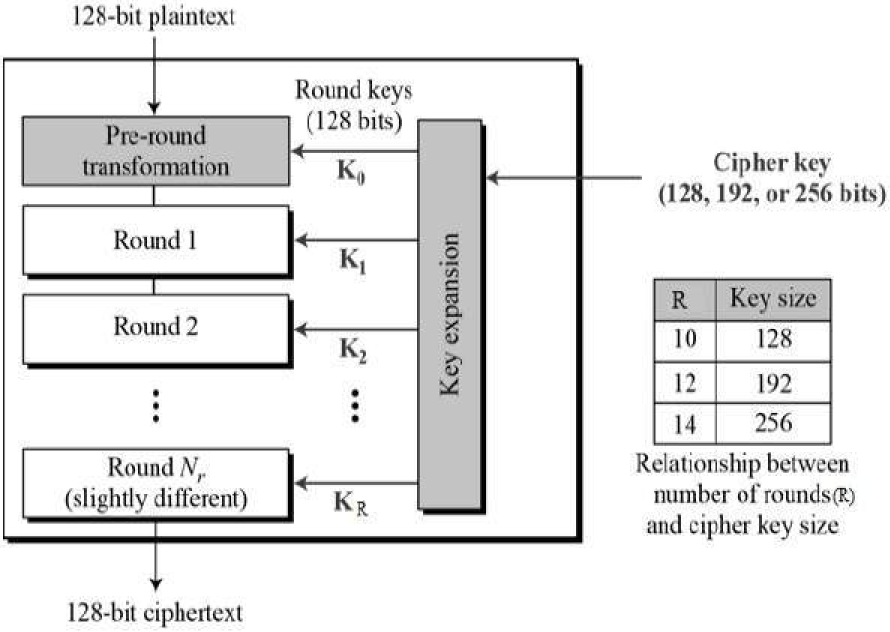


Figure 1.1: Blowfish Block Diagram

1.4 Advanced-Encryption Standard

Rijndael, proposed by Belgian cryptographers, Vincent Rijmen and Joan Daemen, is Symetric-key Block-Cipher that has been established as the Advanced Encryption Standard by the National Institute of Standards and Technology (NIST) of The United States of America, in the year of 2001. The AES may have keys varying in size between 128, 192, 256 bits, & having 10, 12, 14 rounds respectively. The n-Bits key is expanded using AES Key-Scheduling into several subkeys depending on the number of rounds. In the beginning, the input block is XORed with an Initial Round-Key. Then, for the first N-1 rounds, 4 Round Functions are applied on each block. The first-round function is Substitute Bytes where every byte is substituted by another, from the lookup table. Followed by Shift-Rows, where the last three rows are cyclically shifted by certain number of steps. Shift-Rows is followed by Mix-Columns, where a linear mixing operation is executed on the columns, combining the 4-bytes of each column. Lastly, Add-Round-Key function is executed on the current state, where each byte, and a byte of the round key are combined using bit-wise XOR operation. For the Nth round, i.e the last round all the above functions are applied except the Mixed Columns step. AES is one of the most extensively used and secure algorithms for data security. Even though it is slower than blowfish, it provides a higher level of data security.



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Figure 1.2: AES Block Diagram

1.5 Rivest–Shamir–Adleman (RSA) Algorithm

RSA or Rivest–Shamir–Adleman Algorithm, a public-key/asymmetric-key cryptography algorithm that uses a Public Key, available to everyone on the network, to encrypt data and a Private-Key, accessible to only the Sender and Receiver, for decryption. The keys are large prime numbers of lengths 1024 / 2048 / 3072 / 4096 Bits.

Two large prime numbers p and q are selected.

The modulus n is calculated as, $\boldsymbol{n = p \times q}$

Euler's Totient Function of n, $\boldsymbol{\varphi(n) = (p-1) \times (q-1)}$

The Public-key, e is selected, such that e and the Euler's Totient Function of n are co-primes, i.e $\boldsymbol{gcd(e, \varphi(n)) = 1}$

The Private Key, d is calculated such that $\boldsymbol{(d \times e)\ mod\ \varphi(n) = 1}$

Hence the **Public-Key pair is (e , n)** and **Private-Key Pair is (d , n)**.

The Plaintext, M is encrypted by the use of the Public Key, e as:

$$Ciphertext, C = M^e mod\, n$$

The Ciphertext, C is decrypted by use of the Private-Key, d as:

$$Plaintext, M = C^d mod\, n$$

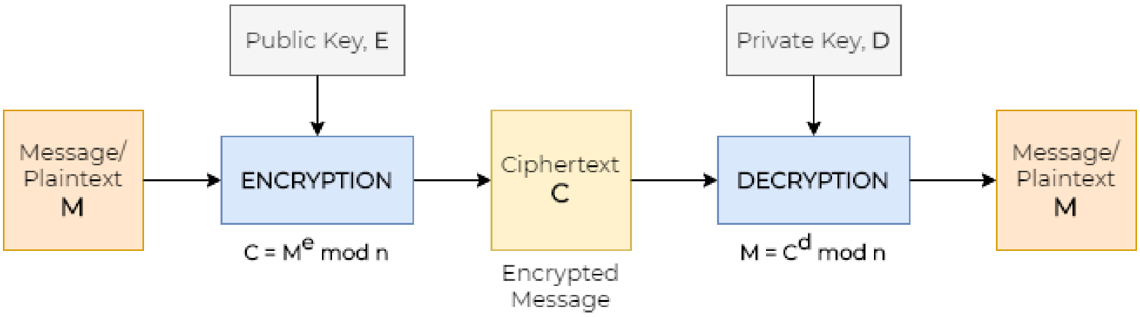


Figure 1.3: RSA Block Diagram

1.6 LSB Image Steganography

Least Significant Bit Steganography is a technique of hiding data within digital media, here, Image. Images are made up of pixels, and the value of each pixel usually refers to the color-code of that pixel. In a photo's gray-scale mode, these pixel values range from 0-255. In LSB Image Steganography, the least-significant bit of a pixel is changed, but that doesn't have much of a visible change in the image. A cover image is where the data is hidden. The cover image is converted to greyscale. The message is converted into binary. Each pixel of the image is traversed through, and for each pixel, initiate a temporary variable, temp. If the LSB of the Pixel Value and the message bit is the same, set temp as 0 and set temp as 1 otherwise. Update the output image pixel as image pixel value added with the temporary variable value, temp. This is done until the message is completely embedded

1.7 SHA-1 Hashing Function

Secure Hashing Algorithm is a one-way hash function that generates a condensed hash of the message, called the message-digest. Any changes made to the message get reflected onto the message-digest, that is if the message changes the message digest will change. This feature of SHA-1 is highly efficient in the generation of random numbers and bits, generation and validation of digital signatures, message authentication codes,

CHAPTER 2

LITERATURE SURVEY

2.1 An Efficient Algorithm for Confidentiality, Integrity, and Authentication Using Hybrid Cryptography and Steganography

Authors, Publication & Year: **Chitra Biswas, Udayan Das Gupta, Md. Mokammel Haque - ICECCE, 2019**

The resistivity of the system proposed by Biswas et al. (2019) , consisting of AES-RSA Data and Key security and LSB Steganography for storing encrypted key, against attacks has been eshtablished. Thus this system provides authentication, integrity and confidentiality together.

2.2 Application of AES & RSA Hybrid Algorithm in E-mail

Authors, Publication & Year: **Ye Liu, Wei Gong, Wenqing Fan - ICIS, 2018**

Liu et al. (2018) showed that Combining asymmetric encryption with symmetric encryption algorithms makes the system significantly secured and faster. The experimental system also shows that Hybrid Crypto-systems are a great alternative to traditional crypto-systems that rely on higher keys sizes and rounds.

2.3 Performance Comparison Between AES256-Blowfish and Blowfish-AES256 Combinations

Authors, Publication & Year: **Muhammad Abdul Muin, Muhammad Abdul Muin, Arief Setyanto, Sudarmawan, Kartika Imam Santoso - ICITACEE, 2018**

The conclusion of the research by Muin et al. (2018) shows that a hybrid cryptosystem on AES256-Blowfish required longer decryption time compared to the one in reverse order (Blowfish-AES256) therefore, is proven to be the most secured one when compared with Blowfish-AES256, Blowfish, or AES256.

2.4 Enhancement the Security of Cloud Computing using Hybrid Cryptography Algorithms

Authors, Publication & Year: **Ali Abdulridha Taha, Dr. Diaa Salama AbdElminaam, Prof. Khalid M Hosny - IJACT, 2017**

Taha et al. (2017) proposes a system that demonstrates that hybrid crypto-systems increase the level of security of mobile data and also reduces the time complexities of encryption and decryption.

2.5 A Research Paper on New Hybrid Cryptography Algorithm

Authors, Publication & Year: **Prof. Swapnil Chaudhari, Mangesh Pahade, Sahil Bhat, Chetan Jadhav, Tejaswini Sawant - IJRDT, 2019**

Swapnil et al. (2018) studies the implementation of a hybrid Crypto-system of symmetric encryption and public-key algorithms. The paper also explores the flows in both standalone systems and uses a hybrid approach to enhance security and address the drawbacks of the standalone systems.

2.6 Performance evaluation of Hybrid Cryptography Algorithm for Secure Sharing of Text and images

Authors, Publication & Year: **Pooja Patil, Dr. Rajesh Bansode - IJRET, 2020**

A combination of AES-ECC and SHA-256 is implemented by Pooja Patil (2020) and targeted towards securing medical sector data. It proves efficient in securing text and image-based data.

2.7 Efficient Hybrid Cryptography Algorithm

Authors, Publication & Year: **Mayes M. Hoobi - Journal of Southwest Jiaotong University, 2020**

Hoobi (2020) proposes A hybrid crypto-system with a combination of DES and ECC, based on test results, that demonstrated to increase the complexity of block cipher.

2.8 Design And Implementation Of A Hybrid Cryptography Textual System

Authors, Publication & Year: **Dr. Mahmood Zaki Abdullah, Zinah Jamal Khaleefah - Transactions on Image Processing (Journal), 2018**

Abdullah and Khaleefah (2018) proposes a system that establishes the use of hybrid crypto-systems increases the level of security of encrypted data and also reduces the time required for encrypting data and decrypting ciphertext.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Problem Identification

Various Encryption Algorithms are used in apps and services to secure data. But the advent of new and sophisticated technologies is making these existing systems obsolete. Advancements in Hardware have significantly reduced the time required to break a cryptographic system. Various kinds of attacks have weakened the existing systems.

Crypto-analysis and special mathematical attacks have made these systems quite vulnerable to being broken by cryptographers. Key security is another vulnerability that modern systems face. Ensuring safe storage and transmission of sensitive keys is a major fault of existing systems.

Another key aspect of securing data is to ensure that performance is not compromised. Generally, encryption algorithms to provide higher levels of security use larger key lengths, but that hampers the performance of the system.

A single layered standalone crypto-system can sometimes have tradeoffs that might lead to data leaks, and also hamper key security. A standalone system has vulnerabilities that often effect the security of data.

The various pitfalls of standalone systems at times compromise the performance and speed. So a requirement of a system that overcomes the performance-security tradeoffs of cryptographic algorithms when used separately is ever more pertinent.

3.2 Proposed Solution

To address the above issues the need for a hybrid approach is higher than ever. The proposed system utilizes a combination of three of the most robust and popular algorithms to secure data.

A combination of Asymmetric Cryptography Algorithm RSA and Symmetric Cryptography Algorithms AES and Blowfish. RSA is one of the most widely used asymmetric encryption algorithms, that is it requires two separate keys to encrypt and decrypt, over the net, specifically on the TLS Layer and used for various other functions apart from encryption of data. Blowfish and AES, on the other hand, are Symmetric Ciphers, that is, it uses identical keys for both encrypting and decrypting data. While Blowfish is the Fastest Encryption algorithm, AES is the most secure and efficient in encrypting data. A combination of these can help in addressing the drawbacks of their standalone counterparts.

The proposed system in this project uses a layered encryption architecture that encrypts data thrice using the three different algorithms and to ensure key security, the keys used are also encrypted and stored in an image using steganography. The keys are encrypted using the hash of the password, as the key for AES. SHA-1 is used to generate the hash from the user input password. The proposed system implemented in Python has proven to be a viable cryptosystem for securing data based on experimental results.

CHAPTER 4

SYSTEM DESIGN

4.1 System Specification

4.1.1 Hardware Specification

The System doesn't require any specialized hardware as it doesn't carry out a large amount of processing. The basic requirements are:

- **Processors:** Intel's Atom® processor or Intel's Core™ processor i3 or above
- **Disk space:** Recommended disk space is 1 GB
- **RAM:** 2GB or more

4.1.2 Software Specification

The project requires an operating system and other software needed for the execution of this application. Operating System provides the underlying environment for the execution of the application. Additional software is needed for the development.

- **Operating systems:** Microsoft Windows 7 or above, Apple's macOS, & Linux
- **Python versions:** Python 3.6.X and above
- **Libraries:** pycrypto, stegano, random and other general purpose libraries

The project is implemented in Python with the help of a few libraries as mentioned next.

4.2 Python Libraries

The libraries used for the project are:

- **PyCrypto**
- **Hashlib**
- **Random**
- **Stegano**

4.2.1 PyCrypto

Python Cryptography Toolkit, also known as PyCrypto is a python library that felicitates the use of various cryptography algorithms and functions required to make a cryptosystem.

It contained several containers and functions in Asymmetric and symmetric cryptography algorithms and hash functions. Several of them are:

- **crypto.PKCS1_OAEP**
- **crypto.Blowfish**
- **crypto.AES**
- **crypto.PublicKey**
- **crypto.PublicKey.RSA**
- **crypto.Util**
- **crypto.Util.Padding**
- **crypto.Util.Padding.pad**
- **crypto.Util.Padding.unpad**

While there are many other functions and containers in the Library, the above are the containers used in the project.

4.2.2 hashlib

It is a hashing library having multiple hash functions and operations. The constructors in the library provide some of the hashing functions used in the project. Some of them are:

- **hashlib.update**
- **hashlib.digest**
- **hashlib.hexdigest**
- **hashlib.sha1**
- **hashlib.sha384**
- **hashlib.sha224**
- **hashlib.md5**
- **hashlib.sha512**
- **hashlib.sha256**

A few of the above functions were used in the project.

4.2.3 stegano

The stegano library is used in the project to facilitate steganography.

- **stegano.lsb**
- **stegano.lsb.hide**
- **stegano.lsb.reveal**
- **stegano.save**

The above was used in the project to facilitate image steganography for the project.

4.2.4 random

The random library is used to generate random numbers and strings. Some of the functions used to generate the ransom strings are:

- **random.randbytes**
- **random.randrange**
- **random.randint**
- **random.choices**
- **random.random**
- **random.gauss**

A few of the above functions were used in the project to generate random strings.

CHAPTER 5

CODING, TESTING

The project is purely implemented in Python based on the following proposed architecture.

The Data or Plaintext is fed into the hybrid system which encrypts the data thrice using three algorithms - Blowfish, RSA, and AES in cascading order.

The keys used are stored in a list that is encrypted using AES, for which the key is generated by hashing a user input Password using SHA1 Hash Function. The system consists of two segments:

- **Data Encryption**
- **Key Encryption**

5.1 Data Encryption

The System consists of three Encryption Layers, a Key Generator, and a List of Keys. The Key Generator generates the random n-bits Key depending on the Encryption Algorithm, while the List of Keys stores the Key Generated in each layer.

Step 1: The plaintext P is first Encrypted using the Blowfish Algorithm with a 32 Bit / 64 Bit / 128 Bit Key, $K_{Blowfish}$. The Key $K_{Blowfish}$ is generated by the Key Generator and is used for Blowfish Encryption. It is then appended to the List of Keys, L. The Plaintext, P is encrypted to generate Ciphertext C_1.

$$C_1 = Blowfish(Plaintext = P, Key = K_{Blowfish}) \tag{5.1}$$

$$L = [\,] \oplus K_{Blowfish} \tag{5.2}$$

Step 2: The Ciphertext, C1 is then encrypted using RSA Encryption with the 1024/2048 Bit Public Key, $K_{RSA-Public}$ generated by the Key Generator. A Private Key, $K_{RSA-Private}$, is also generated for Decryption. While the Public Key is used in Encryption, it is not stored in the List of Keys, L. The Private Key generated is appended to the List of Keys. C_1 is encrypted to generate Ciphertext C_2.

$$C_2 = RSA(Plaintext = C_1, Key = K_{RSA-Public}) \tag{5.3}$$

$$L = [\ K_{Blowfish}\ ] \oplus K_{RSA-Private} \tag{5.4}$$

Step 3: The Ciphertext, C_2 is then encrypted using AES-128 Encryption with the 128 Bit, K_{AES} generated by the Key Generator. The Key, K_{AES} generated is appended to the List of Keys, L. This Step gives the final encrypted ciphertext C.

$$Ciphertext, C = AES(Plaintext = C_2, Key = K_{AES}) \tag{5.5}$$

$$L = [\ K_{Blowfish}\ , K_{RSA-Private}\ ]K_{AES} \tag{5.6}$$

The output of the system is the Ciphertext, C, and the list of keys L with all the keys.

$$List\ of\ Keys, L = [\,K_{Blowfish}\ , K_{RSA-Private}\ , K_{AES}\,] \tag{5.7}$$

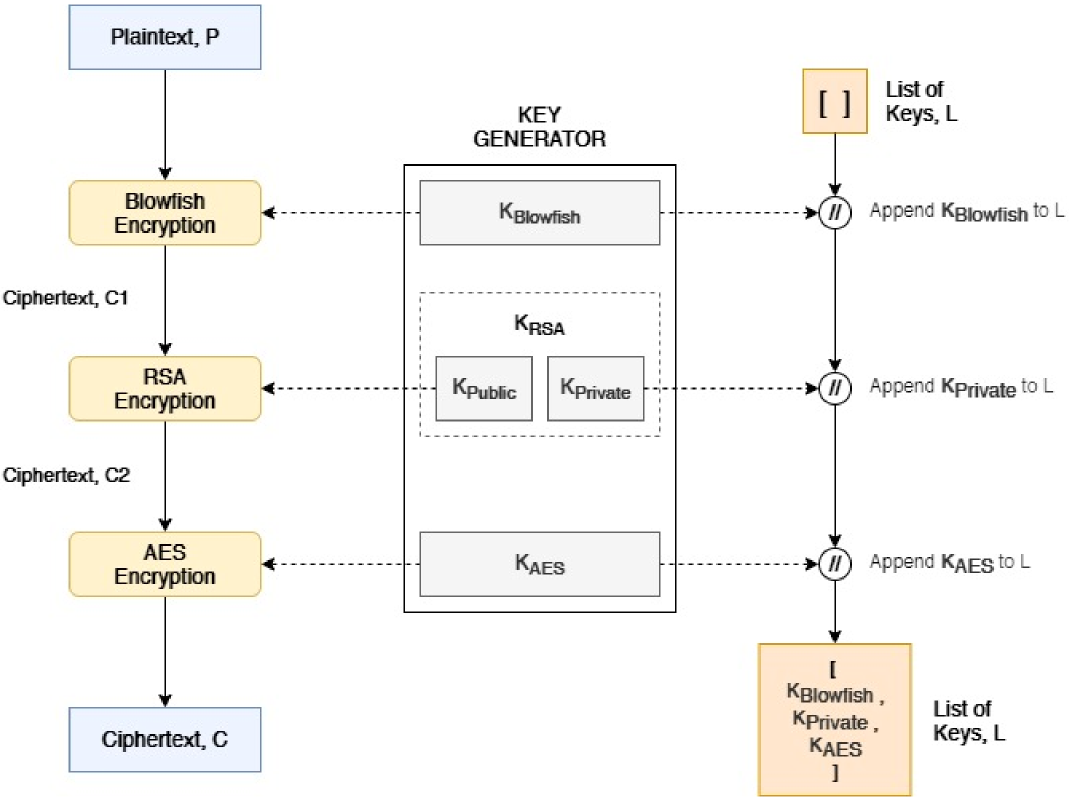


Figure 5.1: Data Encryption Block Diagram

5.2 Key Encryption

Using the proposed system, the Keys used for encryption at the various layers can be securely stored. The List of keys, L stores all the keys generated throughout the Data Encryption Process. Whenever the key for a particular Encryption Layer is generated, it is appended to the List of Keys, L.

In the system, the encryption layers are Blowfish, RSA, and AES, respectively, so the Keys used, are stored in the same order as:

$$List\ of\ Keys, L = [\ K_{Blowfish}\ , K_{RSA-Private}\ , K_{AES}\ ] \tag{5.8}$$

Step 1: This List, L is then passed into a function that converts the list into a single string of keys separated by separators (x , * , /)

$$L_S = Stringify(\ L,\ separator = \ '\times'\ ) \tag{5.9}$$

Step 2: The String, L_S is then encrypted using the AES Encryption Algorithm with a Key generated from user-input password. The user inputs a password, P_W which is hashed using SHA1, & the first 16 Bits of the Hash is used as the key $K_{Password}$. The Key, $K_{Password}$ is used for the Encryption, generating the encrypted string $L_{S-Encrypted}$.

$$HashedPassword, H_P = SHA(P_W) \tag{5.10}$$

$$Key, K_{Password} = H_P[0:16] \tag{5.11}$$

$$L_{S-Encrypted} = AES(L_S, K_{Password}) \tag{5.12}$$

Step 3: This Encrypted string is then Embedded into a Cover Image using Least Significant Bit Steganography, giving the embedded Stego-Image.

$$Stego\ Image = LSB - Steganography(L_{S-Encrypted}\ , Cover - Image) \quad (5.13)$$

The Stego-Image is transferred to the Receiver along with the Encrypted Data.

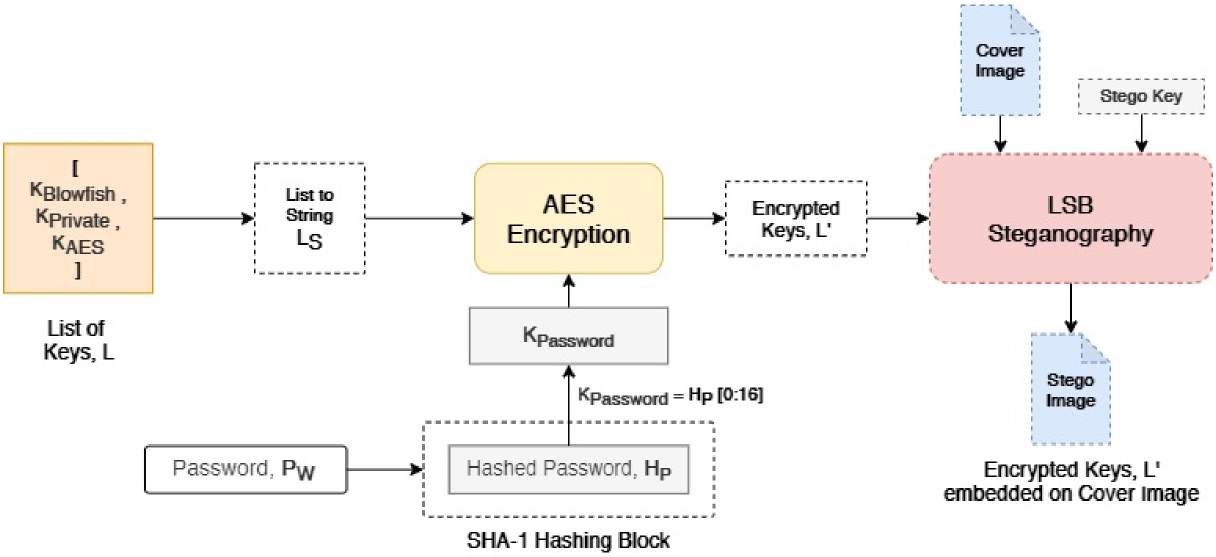


Figure 5.2: Key Encryption Block Diagram

CHAPTER 6

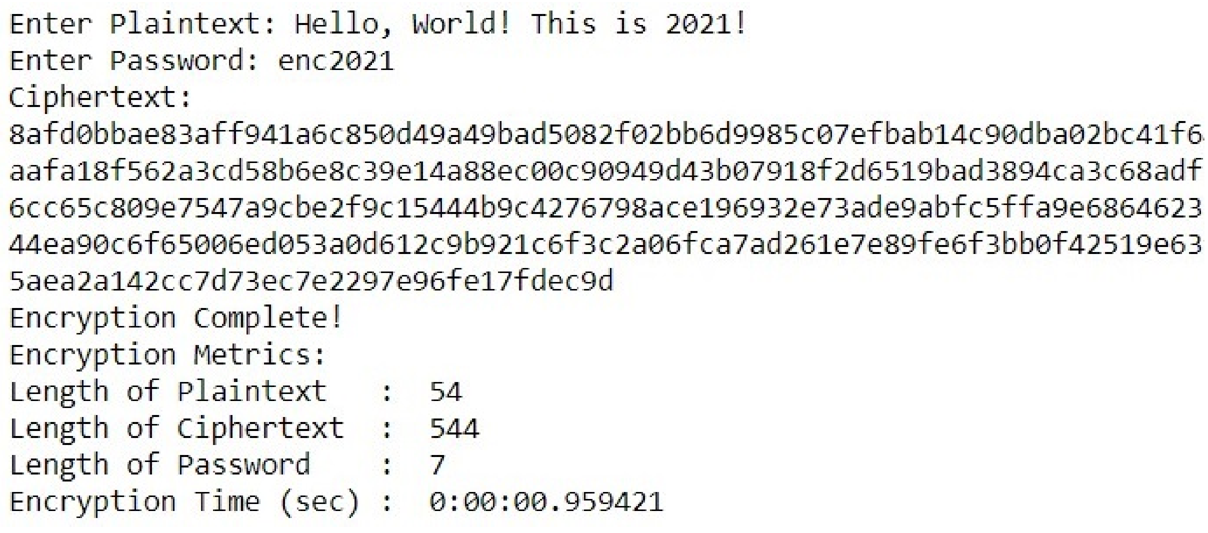
RESULT ANALYSIS

6.1 Experimental Results

The Proposed Hybrid Crypto-system was implemented using Python and tested on a Windows PC with an Intel i3 processor and 4 GB of RAM. For demonstrating the Encryption of data, the following plaintext was encrypted. To encrypt the keys, ***'enc2021'*** was used as the password.

Plaintext: **Hello, World! This is 2021!**

The following results were obtained.



```
Enter Plaintext: Hello, World! This is 2021!
Enter Password: enc2021
Ciphertext:
8afd0bbae83aff941a6c850d49a49bad5082f02bb6d9985c07efbab14c90dba02bc41f6
aafa18f562a3cd58b6e8c39e14a88ec00c90949d43b07918f2d6519bad3894ca3c68adf
6cc65c809e7547a9cbe2f9c15444b9c4276798ace196932e73ade9abfc5ffa9e6864623
44ea90c6f65006ed053a0d612c9b921c6f3c2a06fca7ad261e7e89fe6f3bb0f42519e63
5aea2a142cc7d73ec7e2297e96fe17fdec9d
Encryption Complete!
Encryption Metrics:
Length of Plaintext   :  54
Length of Ciphertext  :  544
Length of Password    :  7
Encryption Time (sec) :  0:00:00.959421
```

Figure 6.1: Output

Results show that the ciphertext is 10 times that of plaintext in length. It can be seen that the plaintext and ciphertext differ by a huge margin and are random, hence we can infer that the system can successfully encrypt the data to a form very different from the original plaintext.

6.2 Performance Analysis of the Proposed System

For performance analysis Files of different sizes and types were encrypted using the proposed system with the same password, for every file. The password used was **“enc2021”**.

Table 6.1: Performance Analysis on various Files

File Type	File Size (in KB)	Encryption Time (in sec)	Decryption Time (in sec)
Markdown File (.md)	7	17.24	1.81
MS-Excel Spreadsheet (.xlsx)	13	19.19	2.14
Portable Network Graphics (.png)	92	18.69	6.28
MS-Word Document (.docx)	118	20.75	7.31
Image File (.jpg)	247	19.06	14.26
MS-PowerPoint Presentation (.pptx)	331	20.09	18.25
RAR Archive (.rar)	658	30.37	34.73
Photoshop Editable (.psd)	745	23.96	41.92
Archive File (.zip)	920	26.65	48.1
Font File (.otf)	4134	88	301
Video File (.mp4)	4980	69	254
Photoshop Editable (.psd)	9544	270	779
Audio File (.mp3)	15475	249	1162

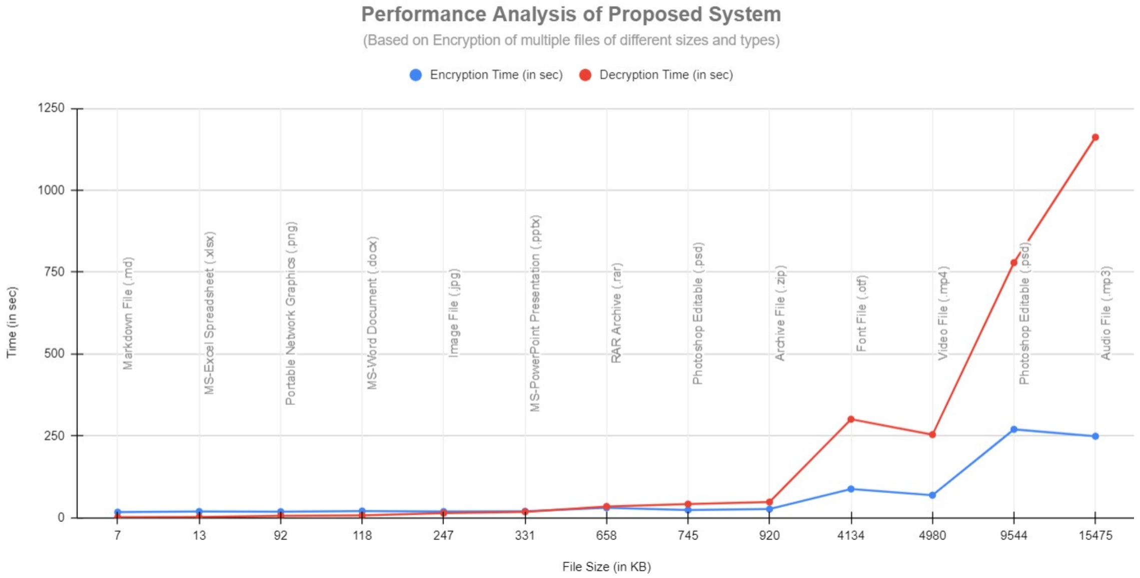


Figure 6.2: Performance Analysis

Table 6.2: Aggregate Performance Analysis

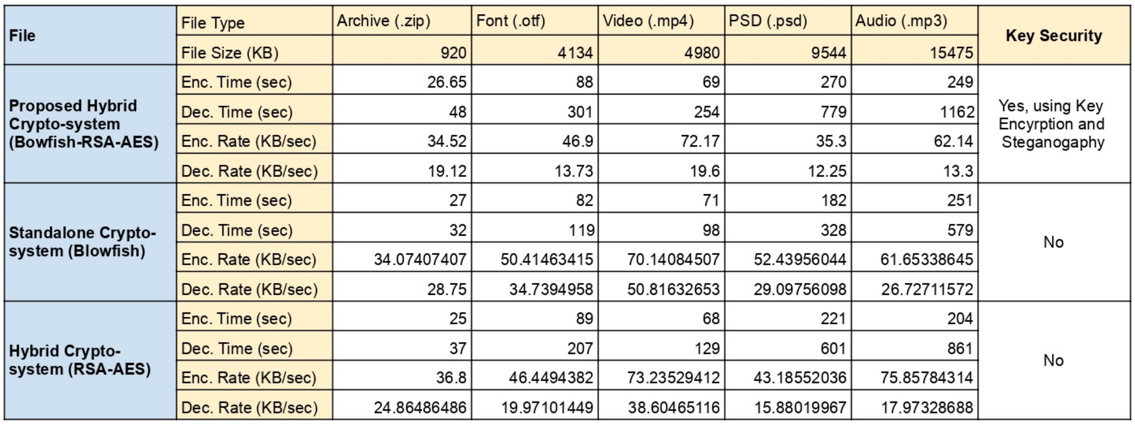
Total Size of Files Encrypted (in KB)	Average Encryption Time (in sec):	Average Decryption Time (in sec):	Average Encryption Rate (in KB/sec):	Average Decryption Rate (in KB/sec):
37264	67.08	205.45	26.54	14.69

On average, we observe that the Rate of Encryption is higher than the Rate of Decryption. Studies have shown that crypto-systems with higher decryption time take more time to break and hence are less susceptible to attacks and more secure. Hence we can conclude that the proposed cryptosystem with high decryption time and rate, is highly secure and efficient in encrypting data.

6.3 Comparison with Existing Systems

Testing two existing systems - one Hybrid (AES-RSA) and one Standalone (Blowfish) with the Proposed Cryptosystem gives the following results.

Table 6.3: Performance Comparison of Three Cryptosystems



File	File Type	Archive (.zip)	Font (.otf)	Video (.mp4)	PSD (.psd)	Audio (.mp3)	Key Security
	File Size (KB)	920	4134	4980	9544	15475	
Proposed Hybrid Crypto-system (Bowfish-RSA-AES)	Enc. Time (sec)	26.65	88	69	270	249	Yes, using Key Encyrption and Steganogaphy
	Dec. Time (sec)	48	301	254	779	1162	
	Enc. Rate (KB/sec)	34.52	46.9	72.17	35.3	62.14	
	Dec. Rate (KB/sec)	19.12	13.73	19.6	12.25	13.3	
Standalone Crypto-system (Blowfish)	Enc. Time (sec)	27	82	71	182	251	No
	Dec. Time (sec)	32	119	98	328	579	
	Enc. Rate (KB/sec)	34.07407407	50.41463415	70.14084507	52.43956044	61.65338645	
	Dec. Rate (KB/sec)	28.75	34.7394958	50.81632653	29.09756098	26.72711572	
Hybrid Crypto-system (RSA-AES)	Enc. Time (sec)	25	89	68	221	204	No
	Dec. Time (sec)	37	207	129	601	861	
	Enc. Rate (KB/sec)	36.8	46.4494382	73.23529412	43.18552036	75.85784314	
	Dec. Rate (KB/sec)	24.86486486	19.97101449	38.60465116	15.88019967	17.97328688	

The Aggregate results of comparison testing is given below. The results of the

Table 6.4: Performance Comparison - Summary

	Average Encryption Time	Average Decryption Time	Average Encryption Rate	Average Decryption Rate
Proposed Hybrid Crypto-system (Blowfish-RSA-AES)	140.53	508.8	50.206	15.6
Standalone Crypto-system (Blowfish)	122.6	231.2	53.74450004	34.0260998
Hybrid Crypto-system (RSA-AES)	121.4	367	55.10561916	23.45880341

Comparison test shows that the Proposed System has a significantly higher Decryption Time, compared to the other two, indicating that the Proposed System is hard to break and will take higher computing power and time to break than the other two, making is highly secured.

CHAPTER 7

CONCLUSION

The proposed cryptosystem uses a combination of symmetric and asymmetric cryptography to secure data. The system also introduces a sub-process to encrypt the keys used for encryption before embedding them in an image. The combination of Blowfish-RSA-AES has significantly improved the security and also ensured that the drawbacks of the standalone systems are addressed. The system also helps in improving security without the use of keys of larger lengths. We have also seen from the test results that the system is less susceptible to brute force attacks as the decryption time is significantly high. The manyfold expansion of plaintext into ciphertext also helps in ensuring a high level of security. While the system successfully does its intended work, it still required minor improvements for larger adoption.

CHAPTER 8

FUTURE ENHANCEMENT

The proposed system is very secure and robust. It has proven to encrypt data and ensure key security. While it is efficient and secure, it was also seen that the encrypted files are generally 2-3 times the size of the original file, hence the encrypted file takes up a significant amount of space to store. This drawback can be addressed by studying it further and making changes to the proposed system. Another improvement that can be made is by making the encryption and decryption time lesser. Further research on the proposed system can also be done by analyzing different order of combinations of the three algorithms used. A slightly different combination can also be studied by replacing one of the algorithms for improved performance.

REFERENCES

1. Abdullah, M. Z. and Khaleefah, Z. J. (2018). "Design and implement of a hybrid cryptography textual system." *2018 International Conference on Engineering and Technology (ICET)*. 1–6.

2. Biswas, C., Gupta, U. D., and Haque, M. M. (2019). "An efficient algorithm for confidentiality, integrity and authentication using hybrid cryptography and steganography." *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. 1–5.

3. Hoobi, M. M. (2020). "Efficient Hybrid Cryptography Algorithm." *Journal of Southwest Jiaotong University*.

4. Liu, Y., Gong, W., and Fan, W. (2018). "Application of aes and rsa hybrid algorithm in e-mail." *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*. 701–703.

5. Muin, M. A., Muin, M. A., Setyanto, A., Sudarmawan, and Santoso, K. I. (2018). "Performance comparison between aes256-blowfish and blowfish-aes256 combinations." *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. 137–141.

6. Pooja Patil, D. R. B. (2020). "Performance Evaluation of Hybrid Cryptography Algorithm for Secure Sharing of Text and Images." *International Research Journal of Engineering and Technology (IRJET)*, 07(09).

7. Swapnil, C., Pahade, M., Bhat, S., Jadhav, C., and Sawant, T. (2018). "A research paper on new hybrid cryptography algorithm.

8. Taha, A., Salama, D.-D., Abdelminaam, S., Khalid, M., and Hosny, K. (2017). "Enhancement the security of cloud computing using hybrid cryptography algorithms." *International Journal of Advancements in Computing Technology*, 9.