International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

Securing Email using Hybrid Encryption System

G.V.S Pavan Mallik[1], Y Saranya Bala[2]

[1,2]B. Tech 3rd year, Dept. of Computer Science and Engineering, Andhra Loyola Institute of Engineering and Technology, Vijayawada-520008, Andhra Pradesh, India.

Abstract – *Email is a vulnerable medium particularly when emails are sent over unsecured or public Wi-Fi networks. Emails can contain crucial information in perspective of tech companies such as project details, budget etc. Hackers who gain access to an email account can access attachments, content such as login credentials, bank account details etc. Encryption has emerged as a key component of these strategies to secure data from malicious outsiders or carelessness of individuals. This paper presents method which upgrades the security of data especially sent through emails by hybrid encryption. The proposed solution is an implementation of Hybrid Encryption that uses RSA and AES cryptographic algorithms. This model encrypts the data in email and can only be decrypted by authorized users.*

***Key Words*: Cryptography, Hybrid Encryption, Decryption, AES, RSA, Hashing, SHA-512.**

1. INTRODUCTION

Electronic mail(e-mail) is a medium of exchanging messages("mails") between people using electronic devices. Emails operate across computer networks, the network here being the Internet. Nowadays mails can be viewed through any electronic device available nearby such as PC, mobile etc. However, storing of these mails is done over cloud and anyone who has the credentials of the user or access to the device can view the sensitive data in the mails thereby causing exposure of data to unauthorised users which may lead to some drastic loss. Here comes the role of cryptography which encrypts the message on sender side and decrypts the message only on recipient side which avoids data exposure.

2. METHODOLOGIES

2.1 Cryptography

Cryptography is the practice and study of techniques for secure communications. Various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography. Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice. Cryptography referred almost exclusively to encryption, which is the process of converting ordinary information (called "plaintext") into unintelligible form (called "ciphertext"). Decryption is the reverse, moving from the unintelligible ciphertext back to plaintext. The detailed operation is controlled both by the "algorithm" and a "key". There are three kinds of cryptography methodologies: symmetric and asymmetric. Hashing is a kind of cryptography.

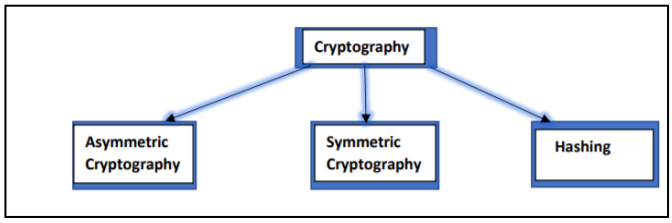

Fig -1: Types of Cryptography

2.1.1 Asymmetric Key Cryptography

In Asymmetric key cryptography, a public key is used to encrypt a message and a private key to decrypt it. This model enhances better security of data. Examples of asymmetric key cryptographic algorithms are RSA (Rivest Shamir and Adleman), Diffie-Hellman.

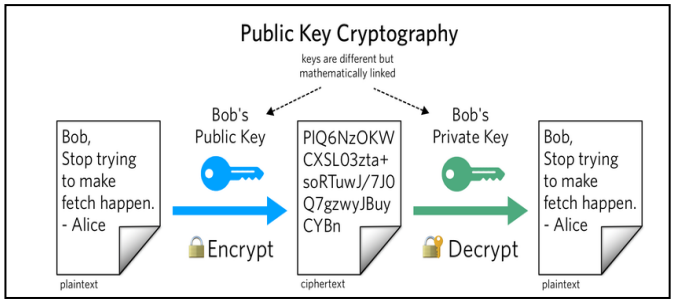

Fig -2: Asymmetric Key Cryptography

2.1.2 Symmetric Key Cryptography

In Symmetric key cryptography, the same secret key is used to both encrypt and decrypt a message. As they use shorter key lengths, the data manipulation is faster. Example of Symmetric key cryptographic algorithms are AES (Advanced Encryption Standard), DES (Data Encryption Standard), 3DES (Triple Data Encryption Standard).

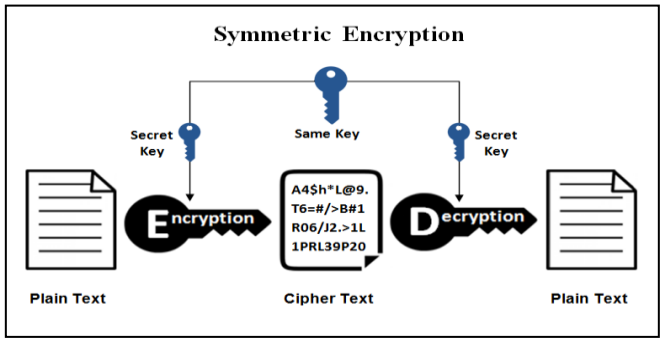

Fig -3: Symmetric Key Encryption

2.1.3 Hashing

Hashing is a method of cryptography that converts any form of data into a unique string of text. Any piece of data can be hashed, no matter its size or type. A hash is designed to act as a one-way function - you put data into a hashing algorithm and get a unique string but if you come upon a new hash, you cannot decipher the input data it represents. There are several hashing algorithms, the main categories are Message digest (MD, MD2, MD4, MD5, MD6), RIPEMD, WHIRLPOOL, Secure Hash Function (SHA-0, SHA-1, SHA-2, SHA-3).

3. ALOGRITHMS

3.1 AES (Advanced Encryption Standard) Algorithm

The most popular and widely used symmetric encryption algorithm is likely to be encountered as Advanced Encryption Standard (AES). AES on the other hand which encrypts all 128 bits in a single iteration. The algorithm uses block encryption of 128 bits in size. It also supports key sizes of 128, 192 and 256 bits.

A round consists of several processing steps to transform input into the final output of ciphertext. AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix.

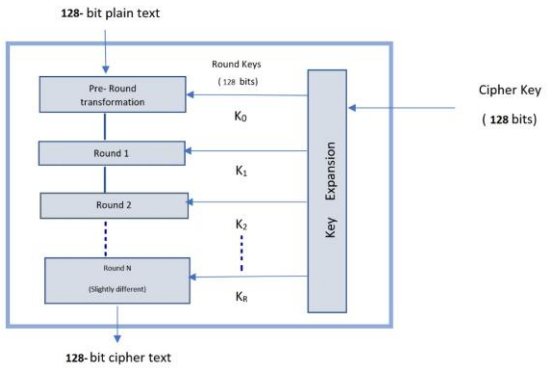

Fig-4: Schematic diagram of AES Structure

Encryption Process:
Each round comprises of four sub-processes.

(1) Byte Substitution (Sub Bytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

(2) Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows –

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

(3) *MixColumns*

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

(4) Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

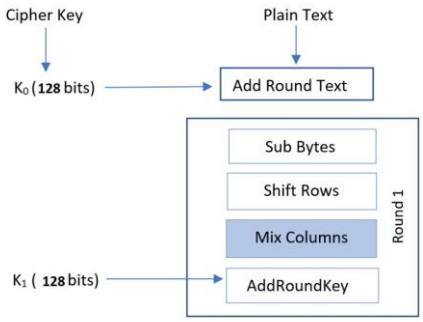

Fig-5: First round of Encryption process

Decryption Process:
The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order.

3.2 RSA (Rivest Shamir and Adleman) Algorithm

The RSA cryptographic algorithm is one of the first public-key cryptosystems, based on math of modular exponentiations and closely related to Integer Factorization

Problem (IFP). The RSA algorithm is named after the initial letters of its authors Rivest-Shamir-Adleman.

The RSA cryptographic algorithm involves 3 phases.

1)Key-Pair Generation
2)Encryption
3)Decryption

Phase 1: Key-Pair Generation

RSA algorithm deals with a public key which is randomly generated and a private key which is related to public key. The public key is used for encryption and private key is used for decryption. The process of key-pair generation is as follows:

(a)Choose two distinct large random prime numbers P and Q.

(b)Compute N = P * Q. The number N is further used as modulus in generating public and private keys.

(c)Compute the Phi of N, ϕ(N) = (P-1) * (Q-1).

(d)Choose an integer E, 1<E< ϕ(N) such that GCD (E, ϕ(N)) == 1; E, ϕ(N) are co-primes. The number E is used as a public key exponent.

(e)Choose D which satisfies the congruence relation, E * D = 1(mod(ϕ(N))). D is used as private key exponent

Public key = (E, N)

Private Key = (D, N)

Phase 2: Encryption

Encryption takes at the sender side. Encryption requires the public key and algorithm.

Ciphertext = (Message ^ E) mod(N)

Phase 3: Decryption

Decryption requires the private key associated the public key used for encryption.

Message = (Ciphertext ^ D) mod(N)

3.3 SHA-512 (Secure Hash Algorithm)

SHA-512 is a hashing algorithm that performs a hashing function on some data given to it. Hashing algorithms are used in many things such as internet security, digital certificates. SHA-512 is a part of a group of hashing algorithms called SHA-2 which included SHA-156 as well which is used in bitcoin blockchain for hashing.

SHA-512 does its work in a few stages. They go as follows:

1) Input Formatting

This includes formatting the given data into a format of N*1024 bits as

<data + padding + size of data>

2) Hash Buffer Initialization

The algorithm works in rounds so the intermediate results should be stored and accessed whenever needed. So, a hash buffer containing 8 registers is maintained called as Initial vector (IV).

3) Message Processing

Message is processed in rounds. Each round it takes the previous processed result and 1024 bits block of the formatted message.

4) Output

After all rounds are completed, we obtain the final 512 bits hash which is the SHA-512 hash of the given data.

4. PROPOSED SOLUTION MODEL

The solution model is used to is used for better security purpose of data sent through emails. This model works among a sender and a receiver. The sender simply encrypts the message he wanted to send using an AES public key and encrypts this AES key using RSA public key of receiver. He also finds a SHA-512 hash value of the original message for validation purpose and appends all this data into a file and sends to receiver through email. The receiver decrypts the message received from sender using his RSA private key. He then validates the hash value received with the hash value he computed for the message he decrypted.

This model has broadly 3 phases. They are as follows

PHASE-1:
SENDER:

(1) Generates AES public key as Keas and RSA encrypts it using receiver's public key as RSA(Keas).

(2) Computes SHA-512 Hash value of the original message as H1(PT1).

(3) AES encrypts of the original message using Keas as AES(PT1).

(4) Appends the text to CipherText.txt as of the format

<RSA(Keas)>
<H1(PT1)>
<AES(PT1)>

PHASE-2:

“”"CipherText.txt is sent from sender to receiver through email"""

PHASE-3:
RECEIVER:

(5) RSA Decrypts of RSA(Keas) using private key of receiver to get AES public key as Keas.

(6) Using Keas, performs AES Decrypt of the message to get original message (PT2).

(7) Computes SHA-512 Hash value of the original message as H2(PT2).

(8) Checks if Hash value sent in file is same as hash value obtained.

IF matches:

original message is successfully decrypted.

ELSE:

message is corrupted.

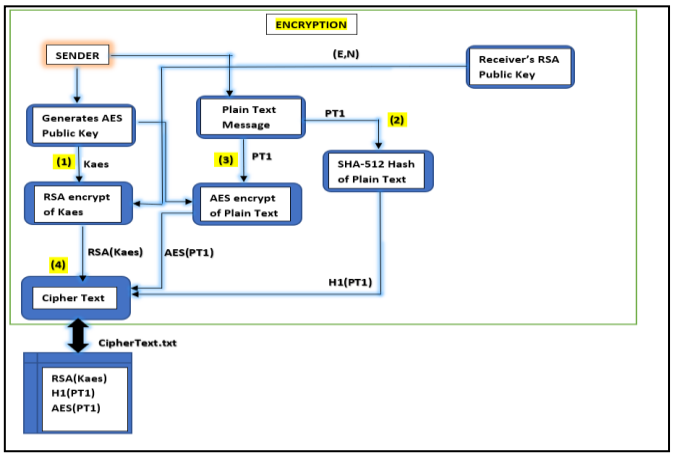

Fig -6: Flow-Chart of Phase-1
Encryption of message

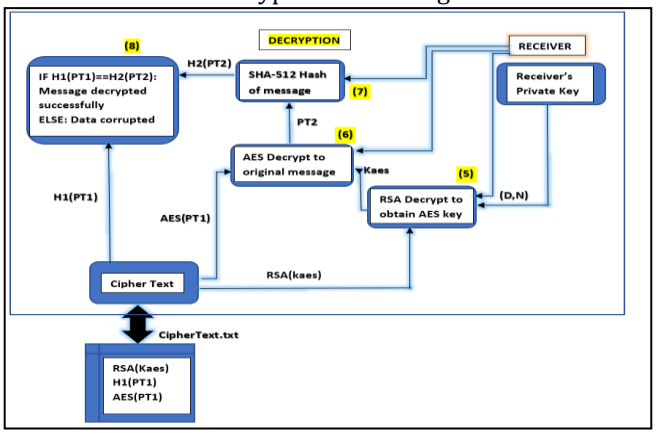

Fig-7: Flow-Chart of Phase-3
Decryption of message and validation

5. IMPLEMENTATION OF MODEL

A Hybrid Encryption Tool has been developed using Python Programming Language, to perform all the phases mentioned in the proposed solution model. To encrypt the message and send to a person, the sender needs to provide his email credentials and public key of receiver. The tool performs all the steps of encryption and sends the final Ciphertext file to receiver over email. To decrypt the message, the receiver has to manually provide each Ciphertext file. The tool performs all decryptions and validates the message. If the message is not corrupted, it displays the original message or else it displays an "Error" message.

```
def encrypt_file(in_file,password):
    with open(in_file, "rb") as fIn:
        with open("cipher.txt.aes", "wb") as fOut:
            pyAesCrypt.encryptStream(fIn, fOut, password, bufferSize)
    # get encrypted file size
    encFileSize = stat("cipher.txt.aes").st_size
    return encFileSize

def decrypt_file(encrypted_file,password,bufferSize,encFileSize):
    with open(encrypted_file, "rb") as fIn:
        with open("output.txt", "wb") as fOut:
            try:
                # decrypt file
                pyAesCrypt.decryptStream(fIn, fOut,password, bufferSize, encFileSize)
            except ValueError:
                print("error")
    print("decryption done")
```

Fig-8: Code Snippet of AES Encryption and Decryption

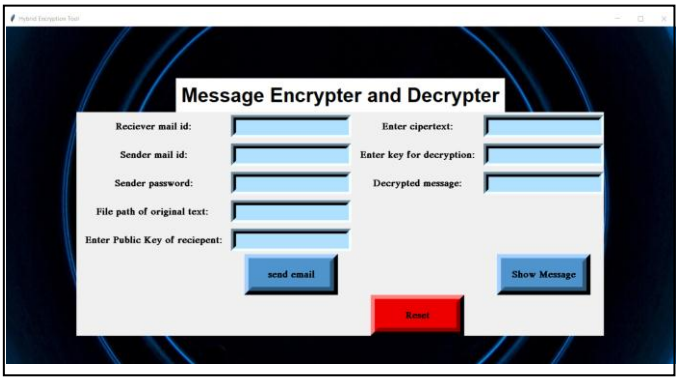

Fig-9: Snapshot of Hybrid Encryption Tool

6. CONCLUSION

Hybrid Encryption System tool uses symmetric and asymmetric key algorithms which provides the confidentiality of the data during transmission. The model is developed by using the combination of RSA algorithm and AES algorithm for making it more secure. The model is expected to be providing more security mechanism in protecting the attachment documents or files for the email services. In a conclusion, the Hybrid Encryption System tool may help users in protecting their confidential files from unauthorized third party while sending the files to the authorized recipients.

REFERENCES

[1] R. L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-key Cryptosystems, Communication of the ACM, vol 21, no 2, (ACM, 1978), pp. 120 – 126.

[2] Neha Tyagi, Ashish Agarwal, "Methods for Protection of Key in Private Key Cryptography", 2017 IEEE, ISSN: 2347- 5552.

[3] https://medium.com/@zaid960928/cryptography-explaining-sha-512-ad896365a0c1

BIOGRAPHIES

G.V.S Pavan Mallik,
B. Tech 3rd year,
Dept. of CSE,
ALIET.

Y Saranya Bala,
B. Tech 3rd year,
Dept. of CSE,
ALIET.