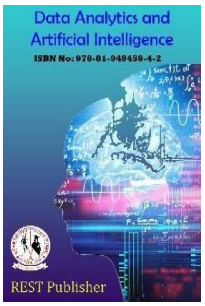
Data Analytics and Artificial Intelligence

Vol: 4(2), 2024

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/4/2/7



A Study on Network Security Through a Combined Cryptographic Strategy

E. Kamalanaban1, *J. Senthil Murugan, S. Sanjay, P. Navien Kumar, V. Parthasarathi

Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Tamil Nadu, India.

*Corresponding Author Email: ***j.senthilmurugan@velhightech.com***

Abstract: *Data security is still one of the most important aspects of our online presence. In order to safeguard data, encryption methods including AES, DES, Blowfish, RSA, and Triple-DES are frequently employed. But as technology develops, traditional approaches to protecting data privacy are losing some of their efficacy. Popular applications like Telegram and WhatsApp lack the modern security features needed to fend off new attackers. Hardware developments have also shortened the time required to crack encryption schemes. A multi-layered encryption system that cascades several sophisticated cryptographic algorithms is therefore becoming increasingly necessary. Using a variety of algorithms, including AES, RSA, and Blowfish, this project seeks to encrypt data while guaranteeing the safe transfer of keys following authentication. This research aims to overcome the weaknesses of standalone data security systems since multi-layered encryption techniques are not routinely applied in existing systems. The proposed approach integrates cryptographic techniques with AI-driven algorithms to fortify the resilience of network infrastructure against emerging threats. AI algorithms will be employed to analyze network traffic patterns, detect anomalies, and identify potential security breaches in real-time. Additionally, IoT devices will be equipped with advanced cryptographic mechanisms for secure authentication and data encryption.*

Keywords: *Cryptography, Least Significant Bit (LSB), Secure Sockets Layer*

1. INTRODUCTION

Cryptography, the study of safeguarding data and communication, aims to protect information from unauthorized access by intruders. Its scope includes creating secure communication protocols, analyzing systems for weaknesses, and ensuring integrity, authenticity, confidentiality, and non-repudiation. It forms the foundation of information security, crucial for maintaining one's digital presence. Encryption, a central aspect of cryptography, transforms readable data into unreadable text using cryptographic algorithms or systems. Modern systems heavily rely on mathematical theories and computational practices. Encryption involves mathematical functions requiring a secret key known to the user. Decryption, on the other hand, reverses this process, converting encrypted data back to its original form using the key. Cryptography algorithms fall into two main categories: Symmetric-Key Cryptography and Asymmetric Cryptography. Symmetric-Key systems use the same key for both encryption and decryption, seen in algorithms like AES and Blowfish. In contrast, Asymmetric Cryptography, or Public-Key encryption, utilizes different keys—a public one for encryption and a private one for decryption. In response to these challenges, this mini project proposes a novel approach to network security by combining cryptographic strategies with AI and IoT technologies. By integrating these three pillars, the aim is to develop a comprehensive framework that not only protects against known threats but also adapts to emerging security risks in real-time.

2. SYSTEM SPECIFICATION

Hardware specifications:

- •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

Software specifications:

- •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

3. METHODOLOGY

1.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.
2.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 Stan-dard by the National Institute of Standards and Technology (NIST) of The United Statesof 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 us-ing AES KeyScheduling 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 firstN-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. Fol-lowed by Shift-Rows, where the last three rows are cyclically shifted by certain numberof 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 roundkey are combined using bit-wise XOR operation.
3.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, $n = p \times q$ Euler's Totient Function of n, $\phi(n) = (p-1) \times (q-1)$ The Public-key, e is selected, such that e and the Euler's Totient Function of n are coprimes, i.e $\gcd(e, \phi(n)) = 1$ The Private Key, d is calculated such that $(d \times e) \bmod \phi(n) = 1$ Hence the Public-Key pair is (e , n) and Private-Key Pair is (d , n).

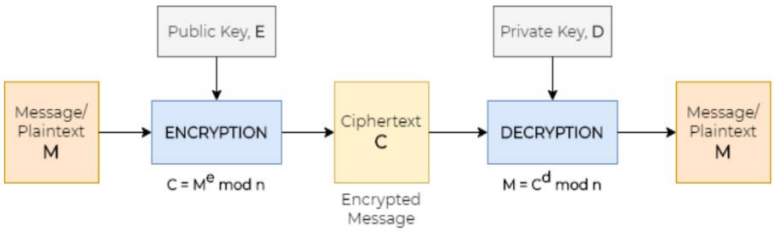


FIGURE 1. RSA Block Diagram

4.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 valuesrange 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

5.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

4. 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 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.

Proposed Solution:

To address the above issues the need for a hybrid approach is higher than ever. The pro- posed system utilizes a combination of three of the most robust and popular algorithms 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

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

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, KBlowfish. The Key KBlowfish 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 C1

C1 = Blowfish(Plaintext = P, Key = KBlowfish) (5.1)

L = [ ] ⊕ KBlowfish

Step 2: The Ciphertext, C1 is then encrypted using RSA Encryption with the 1024/2048 Bit Public Key, KRSA−Public generated by the Key Generator. A Private Key, KRSA−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. C1 is encrypted to generate Ciphertext C2.

C2 = RSA(Plaintext = C1, Key = KRSA−Public) (5.3)

L = [KBlowfish] ⊕ KRSA−Private (5.4)

Step 3: The Ciphertext, C2 is then encrypted using AES-128 Encryption with the 128 Bit, KAES generated by the Key Generator. The Key, KAES generated is appended to the List of Keys, L. This Step gives the final encrypted ciphertext C.

Ciphertext, C = AES(Plaintext = C2, Key = KAES) (5.5)

L = [KBlowfish , KRSA−Private]KAES (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 = [KBlowfish , KRSA−Private , KAES] (5.7)

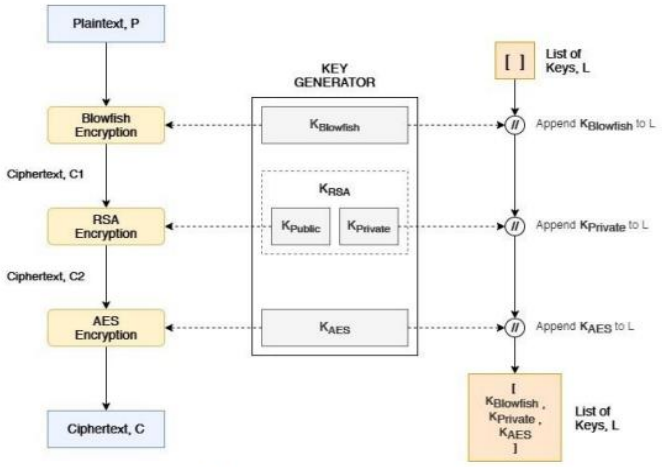


Figure 5.1: Data Encryption Block Diagram

FIGURE 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 = [KBlowfish , KRSA−Private , KAES] (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 , * , /)
LS = Stringify(L, separator = J× J) (5.9)
Step 2: The String, LS is then encrypted using the AES Encryption Algorithm with a Key generated from user-input password. The user inputs a password, PW which is hashed using SHA1, & the first 16 Bits of the Hash is used as the key KPassword .
The Key, KPassword is used for the Encryption, generating the encrypted string LS−Encrypted.
HashedPassword, HP = SHA(PW) (5.10)
Key, KPassword = HP [0 : 16] (5.11)
LS−Encrypted = AES(LS, KPassword) (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(LS−Encrypted , Cover − Image) (5.13)
The Stego-Image is transferred to the Receiver along with the Encrypted Data.

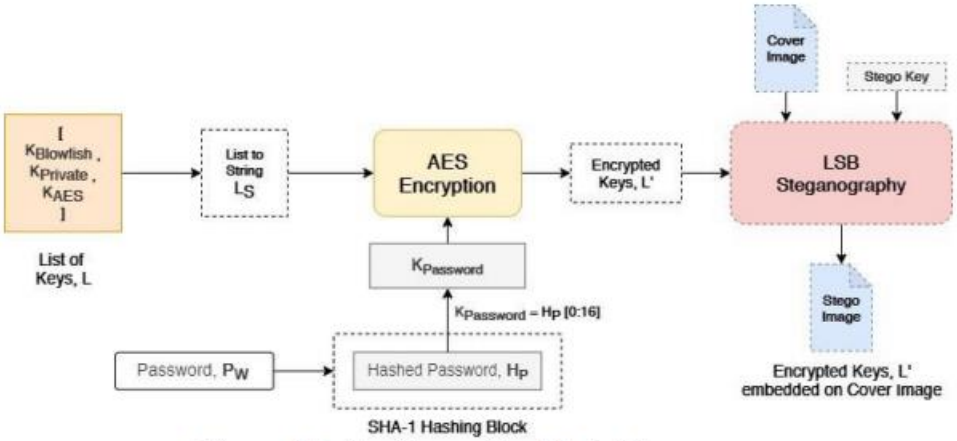


Figure 5.2: Key Encryption Block Diagram

FIGURE 3.

3. 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 theming an image. The combination of Blowfish-RSAAES 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.

REFERENCES

[1]. V. Adat and B. Gupta, "Security in Internet of Things: issues, challenges, taxonomy, and architecture", Telecommunication Systems, vol. 67, no. 3, pp. 423- 441, 2017. Available: 10.1007/s11235-017-0345-9 [Accessed 8 October 2021].

[2]. M. Saleh and H. Hashim, "HYBRID CRYPTOGRAPHIC APPROACH FOR INTERNET OF THINGS APPLICATIONS: A REVIEW", Journal of Information and Communication Technology, vol. 19, no. 3, pp. 279-319, 2020. Available: 10.32890/jict2020.19.3.1.

[3]. L. Ma, X. Sun and W. Jin, "Symmetric–asymmetric hybrid encryption and decryption system based on chaotic iris phase mask and computer-generated holography", Optical Engineering, vol. 59, no. 08, 2020. Available: 10.1117/1.oe.59.8.083106.

[4]. V. Panwar, D. Kumar Sharma, K. Pradeep Kumar, A. Jain and C. Thakar, "Experimental investigations and optimization of surface roughness in turning of en 36 alloy steel using response surface methodology and genetic algorithm", Materials Today: Proceedings, 2021. Available: 10.1016/j.matpr.2021.03.642

[5]. S. Bellovin and M. Merritt, "An attack on the Interlock Protocol when used for authentication", IEEE Transactions on Information Theory, vol. 40, no. 1, pp. 273-275, 1994. Available: 10.1109/18.272497.

[6]. S. Bahtiyar, "A Hybrid Trust-Modeling Approach for IoT Security", Electrica, vol. 20, no. 1, pp. 86-96, 2021. Available: 10.5152/electrica.2021.19090.

[7]. H. Sedjelmaci and M. Feham, "Novel Hybrid Intrusion Detection System For Clustered Wireless Sensor Network", International Journal of Network Security & Its Applications, vol. 3, no. 4, pp. 1- 14, 2021. Available: 10.5121/ijnsa.2021.3401.

[8]. R. Kavitha and B. Caroline, "Hybrid Energy-Efficient Transmission Protocol for Heterogeneous Wireless Sensor Networks", Circuits and Systems, vol. 07, no. 06, pp. 897-906, 2021. Available: 10.4236/cs.2021.76077.