Research Paper: Wannahide

Abdullah Riad, Fouad Mostafa, Mahmoud Mohammed, Abdelrahman Fathy

Misr University for Science & Technology

Abstract—It is generally noticeable that the digital communication whether it involves instant messaging, voice and video calls has became a daily need all over the globe thus protecting any type of data in this type of communication imposed new challenges to the security field which lead to utilizing encryption algorithms to ensure the confidentiality of the data along with its integrity and availability, since ensuring the medium was the first step to achieve that another level of security was added by encrypting the original image into a meaningless cipher using a hybrid encryption-scheme based on AES-256-CBC to secure its content using both symmetric and asymmetric encryption making sure the meaningful image is delivered.

Keywords: ElGamal-Algorithm Double-Ratchet Symmetric/Asymmetric-encryption AES-256-CBC Hybrid-encryption

1- INTRODUCTION

The increasing frequency and severity of data breaches in recent years have highlighted the critical importance of security and privacy in today's digital world. From financial institutions to social media platforms, no organization is immune to the risk of data breaches, which can lead to significant financial losses and reputational damage. This has led to a growing demand for secure communication channels, particularly in light of recent privacy scandals involving social media platforms. The need for a secure chat app that prioritizes user privacy and security has become increasingly evident. A secure chat app can provide end-to-end encryption, preventing unauthorized access to conversations, while ensuring the privacy of the users. In addition, such an app can offer features such as anonymous chat and secure file sharing, giving users control over their data and minimizing the risk of data breaches. The development of a secure chat app requires a comprehensive approach that includes risk assessment, threat analysis, and adherence to industry standards and best practices. By prioritizing security and privacy in the design and development of a chat app, individuals and organizations can communicate safely and confidently, without fear of data breaches or unauthorized access.

Furthermore the need for secure communication and collaboration in various contexts, such as business, government, and personal settings was obvious. With the rise of cyber threats and data breaches, there is a growing concern about the privacy and security of sensitive information transmitted through traditional communication channels, such as email and instant messaging. This problem is compounded by the fact that many existing communication tools lack sufficient encryption and security features to protect against these threats such as data leaks and breaches, which can result in the loss or exposure of sensitive or confidential information since it can occur through a variety of channels, including email, file sharing, messaging applications, and cloud storage, and can have serious consequences for individuals and organizations alike, including financial losses, legal liabilities, and damage to reputation

By providing a practical and effective solution to protect messages transmission that aims to raise awareness of the importance of secure communication it encourages individuals and organizations to adopt more secure messaging tools to protect their sensitive information from unauthorized access and interception.

2- Methodologies:

2.1 ElGamal encryption algorithm:

It is a public-key crypto-system that allows secure communication over insecure channels by encrypting messages using the recipient's public key. The algorithm consists of two main parts: key generation and encryption/decryption.

In the key generation process, the sender (Alice) generates a pair of keys: a private key and a public key. The private key is kept secret and used to decrypt messages, while the public key is shared with others and used to encrypt messages. The key generation process involves the following steps:

1. Choose a large prime number p and a generator g of the multiplicative group of integers modulo p.

2. Choose a random secret number a, such that $1 < a < p - 1$.

3. Compute $A = g^a \bmod p$ and publish (p, g, A) as the public key.

Keep a as the private key.

In the encryption process, the sender (Bob) encrypts a message M using Alice's public key (p, g, A) as follows:

1. Choose a random secret number k, such that $1 < k < p - 1$ and $\gcd(k, p - 1) = 1$.

2. Compute $B = g^k \bmod p$ and $C = AM^k \bmod p$.

3. Send (B, C) as the encrypted message.

In the decryption process, Alice uses her private key a to decrypt the message (B, C) as follows:

1. Compute D = B^a mod p.

2. Compute M = CD^-1 mod p.

As has been demonstrated its based on the computational hardness of the discrete logarithm problem, which is the difficulty of computing the discrete logarithm of a given number modulo a prime. The security of the algorithm depends on the difficulty of computing the private key a from the public key (p, g, A) and the encrypted message (B, C). If the key size and other parameters are chosen appropriately.

2.2 The double ratchet protocol:

It is a cryptographic technique used to secure communication between two parties, such as in messaging applications. The protocol utilizes the double ratchet model to provide forward secrecy and message integrity.

In this protocol, both parties generate a pair of public and private keys. The public key is shared between the parties and used for encryption, while the private key is kept secret and used for decryption. The parties also generate a shared secret key that is used to encrypt and decrypt messages.

The double ratchet model is then used to establish a session key that is used to encrypt messages. The first ratchet is used to generate a new key for each message, while the second ratchet is used to update the shared secret key.

When a message is sent, the sender generates a new message key, denoted as mk, and encrypts the message using the session key derived from the shared secret key and the message key. This is represented by the following equation:

- C = E(Sk, E(mk, M))

where C is the ciphertext, Sk is the session key derived from the shared secret key, E is the encryption function, mk is the message key, and M is the message.

The session key, Sk, used in the double ratchet protocol is derived from the shared secret key, SS, and a message key, mk, as follows:

- Sk = HKDF(SS, mk)

The sender then uses the first ratchet to generate a new message key, denoted as mk', which is used for the next message. This is represented by the following equation:

- mk' = HKDF(ck, 1)

where HKDF is a key derivation function and ck is the chaining key.

The receiver then uses the second ratchet to update the shared secret key. The receiver first decrypts the ciphertext using the session key derived from the shared secret key and the message key, and then uses the decrypted message key to update the shared secret key. This is represented by the following equations:

- mk = E(Sk, E(mk', M'))
- Sk' = HKDF(Sk, ck)

where M' is the decrypted message, Sk' is the updated session key derived from the updated shared secret key, and ck is the updated chaining key.

The sender and receiver then repeat this process for each subsequent message, generating new message keys and updating the shared secret key, ensuring forward secrecy and message integrity.

To sum up , the double ratchet end-to-end encryption protocol utilizes the double ratchet model to provide secure communication between two parties. The protocol provides forward secrecy and message integrity by generating a new message key for each message and updating the shared secret key using the double ratchet model. This protocol provides a secure communication channel for messaging applications and is widely used in modern end-to-end encrypted messaging services.

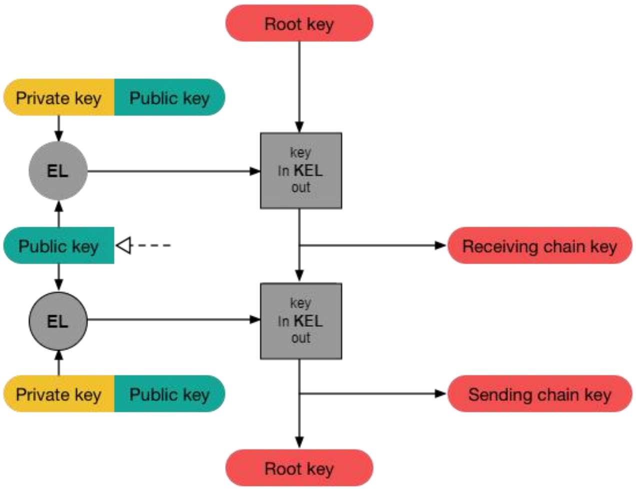

Figure 1. Double Ratchet Mechanism

2.3 AES-256-CBC:

AES-256-CBC refers to a specific encryption algorithm and mode of operation used for symmetric encryption. Let's break down its components:

AES: AES stands for Advanced Encryption Standard, which is a widely used symmetric encryption algorithm. It was selected by the U.S. National Institute of Standards and Technology (NIST) as a replacement for the older Data Encryption Standard (DES). AES operates on fixed-size blocks of data, typically 128 bits, and supports key sizes of 128, 192, and 256 bits.

256: The number 256 indicates the key size used in AES-256. It means that the encryption algorithm employs a 256-bit key for encrypting and decrypting data. AES-256 is considered highly secure and suitable for protecting sensitive information.

CBC: CBC stands for Cipher Block Chaining, which is a mode of operation for block ciphers like AES. In CBC mode, each block of plain-text is combined with the previous cipher-text block before encryption. This chaining process adds randomness and increases the security of the encryption. CBC requires an initialization vector (IV) that serves as the initial input to the encryption algorithm.

To encrypt data using AES-256-CBC:

1. Generate a 256-bit encryption key.
2. Generate a random 128-bit initialization vector (IV).
3. Divide the plaintext into fixed-size blocks (usually 128 bits).
4. XOR (exclusive OR) the first plaintext block with the IV.
5. Encrypt the XOR result using AES-256 with the encryption key.
6. XOR the resulting ciphertext block with the next plaintext block.
7. Encrypt the XOR result using AES-256 with the encryption key.
8. Repeat the XOR and encryption steps for each subsequent block, using the previous ciphertext block in the XOR operation.
9. The final output is the encrypted ciphertext.

To decrypt data using AES-256-CBC, the process is reversed:

1. Retrieve the encryption key and IV.
2. Divide the ciphertext into fixed-size blocks.
3. Decrypt the first ciphertext block using AES-256 and the encryption key.
4. XOR the decrypted block with the IV (for the first block) or the previous ciphertext block (for subsequent blocks).
5. Decrypt the next ciphertext block using AES-256 and the encryption key.
6. XOR the decrypted block with the previous ciphertext block.
7. Repeat the XOR and decryption steps for each subsequent block.
8. The final output is the decrypted plaintext.

To sum up, AES-256-CBC is a widely used encryption algorithm that provides strong security for data at rest and in transit. It is a symmetric-key block cipher that uses a fixed-length key of 256 bits to encrypt and decrypt data. In CBC mode, the algorithm uses a block cipher to encrypt plain-text in blocks of fixed size and applies an initialization vector (IV) to the first block to increase randomness and prevent patterns from being detected in the encrypted data.

The security of AES-256-cbc encryption depends on the strength of the symmetric key and the randomness of the initialization vector. The key size of 256 bits provides a very large key space, making brute-force attacks impractical. The use of a random initialization vector ensures that each block of cipher-text is unique, making it difficult to detect patterns in the encrypted data.

2.4 Hybrid encryption:

It is a cryptographic technique that combines the strengths of symmetric encryption (AES-256-CBC) and public-key encryption (Elgamal cryptosystem). It is computationally efficient and provides strong security guarantees. This approach provides the efficiency of symmetric encryption with the added security of public-key encryption.

To perform a hybrid encryption scheme the following steps take place:

1. Key Generation: The sender generates a symmetric key, K, and a public-private key pair, (PK, SK), using ElGamal algorithm.

2. Symmetric Encryption: The sender encrypts the text, P, using the symmetric key, K, and a symmetric encryption algorithm, E, to obtain the cipher-text, C.

- C = E(K, P)

3. Public-Key Encryption: The sender encrypts the symmetric key, K, using the recipient's public key, PK, and a public-key encryption algorithm, F, to obtain the encrypted key, K'.

- K' = F(PK, K)

4. Transmission: The sender transmits the encrypted text, C, and the encrypted key, K', to the recipient.

Decryption: The recipient receives the encrypted text, C, and the encrypted key, K', and performs the following steps to decrypt the text:

1. Private-Key Decryption: The recipient decrypts the encrypted key, K', using their private key, SK, and a private-key decryption algorithm, G, to obtain the symmetric key, K.

- K = G(SK, K')

2. Symmetric Decryption: The recipient decrypts the encrypted text, C, using the symmetric key, K, and the same symmetric encryption algorithm, E, to obtain the original text, P.

- P = E(K, C)

2.5 MongoDB:

It is a popular open-source NoSQL document-oriented database that provides high-performance, high-availability, and automatic scaling features. It is designed to handle large and complex data sets, and is often used in modern web applications, data analytics, and mobile applications.

Unlike traditional relational databases, MongoDB stores data in documents, which are JSON-like structures that can have nested fields and arrays. Each document can have a unique identifier, called the Object-Id, and can be indexed for fast retrieval. MongoDB uses a flexible schema, allowing data to be added or removed without the need for predefined tables and columns.

One of the key features of MongoDB is its scalability. It supports horizontal scaling through sharding, which is the process of splitting data across multiple servers or shards. Sharding allows MongoDB to handle large data sets by distributing the load across multiple machines, which increases performance and availability. In addition, MongoDB provides automatic failover and replica sets, which ensure high availability and fault tolerance.

MongoDB also provides a rich set of querying capabilities, allowing users to search for data using a variety of operators, such as comparison, logical, and regular expression operators. Queries can be optimized using indexes, which improve performance by reducing the number of documents that need to be scanned.

Overall, MongoDB provides a flexible, scalable, and high-performance database solution for modern applications. Its document-oriented approach and support for horizontal scaling make it well-suited for handling large and complex data sets, while its flexible schema and rich querying capabilities make it easy to work with and adapt to changing requirements.

3- Proposed System:

WannaHide is a secure chatting app that uses hybrid encryption to provide for its users an end-to-end encryption through using Elgamal algorithm for encrypting a symmetric key to provide secure key exchange, then the message is encrypted by this key using AES-256-CBC encryption scheme, in addition the key is passed to the double ratchet protocol that guarantees that each messages is encrypted using different key from the previous and the following messages.

3.1 System Architecture:

The following system architecture shows the life cycle of a message as it starts from A all the way to B safe and secure.

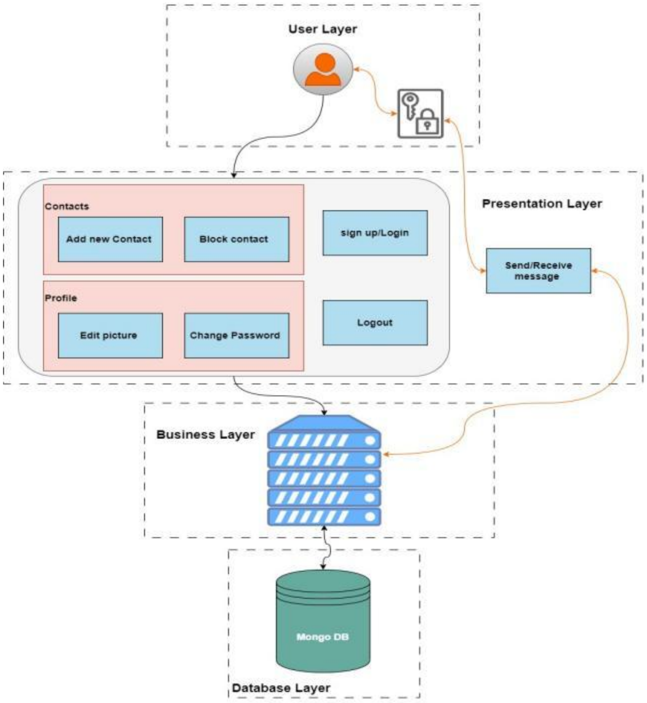

Figure 2. System Architecture

3.2 Encryption scheme of proposed system Wannahide:

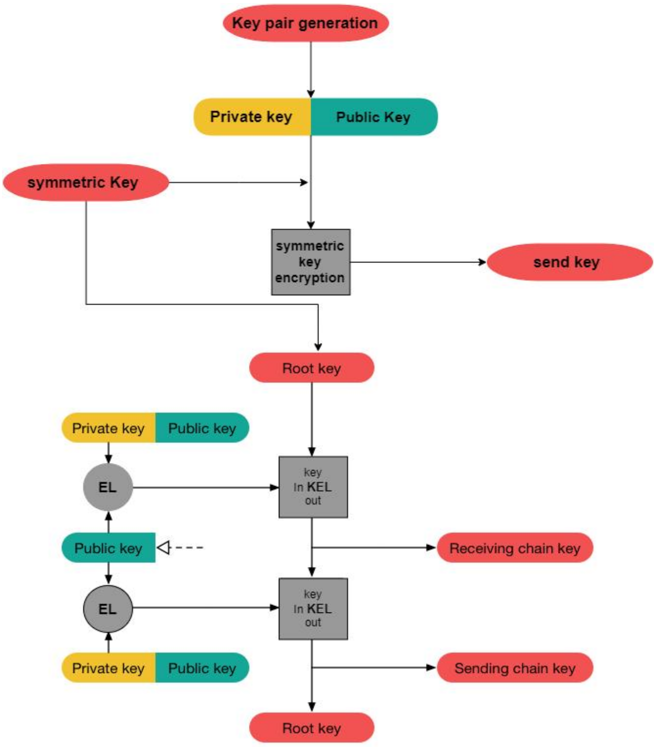

Figure 3. WannaHide encryption mechanism

4- Results:

RSA and ElGamal cryptographic algorithms were implemented in JavaScript programming language on mixed data (text, image). The experimental results of each data set are indicated using tables and figures. As shown bellow, the following tables and figures give the time taken to encrypt and decrypt each data set are given in seconds (s).

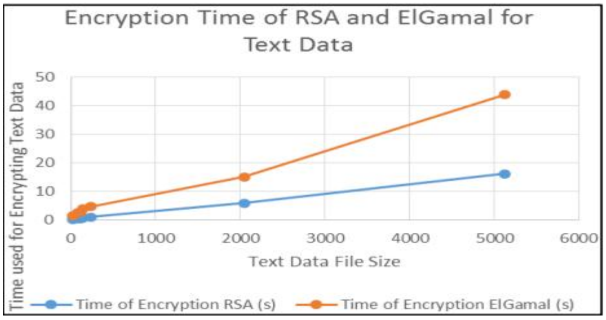

Figure 4. Encryption time analysis for RSA and ElGamal cryptographic algorithms for text data set.

S/N	Text data file size (KB)	RSA(s)	Elgamal(s)
1	22	0.1082	1.55
2	80	0.3545	2.57
3	120	0.4835	2.92
4	140	0.5664	3.80
5	230	0.9315	4.67

Table 1. Tabular representation of text data encryption for RSA and ElGamal algorithms

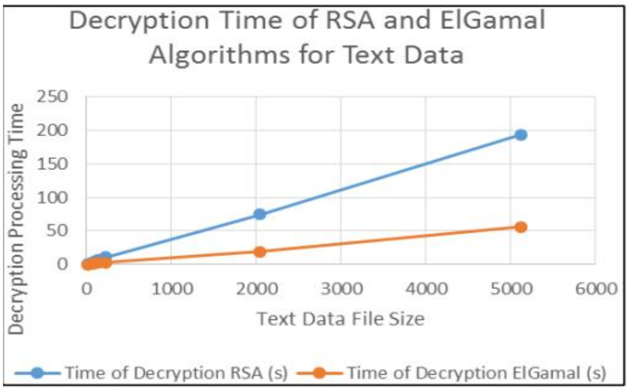

Figure 5: Decryption time analysis of RSA and ElGamal algorithms for text data set

S/N	Text data file size (KB)	RSA(s)	Elgamal(s)
1	22	1.0756	0.0802
2	80	3.9254	1.6674
3	120	5.7463	1.9284
4	140	6.8078	2.2112
5	230	11.1189	3.2596

Table 2. Tabular representation of text data decryption for RSA and ElGamal algorithms

As for images encryption, we convert the image data into base64 string to perform encryption and decryption processes, then regather the string to build up the image again on the receiver side.

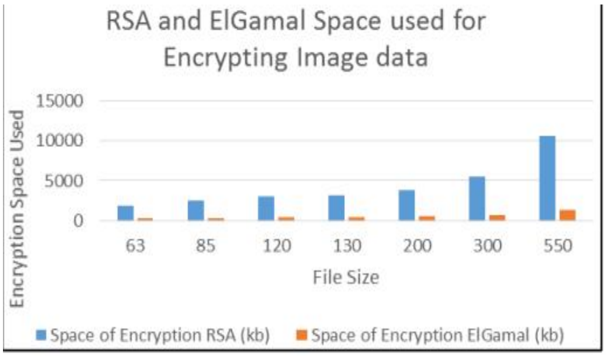

Figure 6: RSA and ElGamal storage space occupied for encrypting image

S/N	Text data file size (KB)	RSA(s)	Elgamal(s)
1	63	0.9896	2.9947
2	85	1.0023	3.3907
3	120	1.6205	8.7705
4	130	1.7495	9.3232
5	300	1.9853	10.5232

Table 3.Tabular representation of image encryption for RSA and ElGamal algorithms

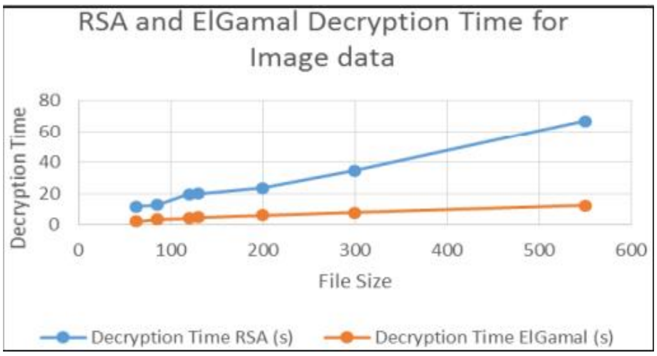

Figure 7: RSA and ElGamal decryption time for images

S/N	Text data file size (KB)	RSA(s)	Elgamal(s)
1	63	11.8935	2.4517
2	85	12.6888	3.8033
3	120	19.6372	4.3965
4	130	19.9276	4.9207
5	300	23.6912	6.3696

Figure 8. Tabular representation of images decryption for RSA and ElGamal algorithms

- Applying Elgamal Algorithm with double ratchet protocol as ehanced alternative of Diffie-Hellman.
- Users are able to exchange messages safe and secure.
- Encryption/Decryption processes are done with no flaws and delivered the desired results.

- The experimental results showed that the RSA algorithm performs better in time complexity for all categories of the data set (text, image) during the encryption process.

- Elgamal algorithm performs better in terms of memory consumption for encryption and decryption processes for all the data set categories.

- Elgamal algorithm has proven to be more secure and challenging to solve than the RSA algorithm because ElGamal has a complicated calculation to solve discrete logarithms.

5- Conclusion :

Purpose of Wannahide is to make instant messaging service more secure through double ratchet end-to-end encryption based on an enhanced algorithm of Diffie-hellman called Elgamal algorithm that made encryption process more secure, faster and less vulnerable to practical attacks that threats the confidentiality of the communication process.

6- References:

[1] Mallouli, F., Hellal, A., Saeed, N. S., & Alzahrani, F. A. (2021, June). A survey on cryptography: comparative study between RSA vs ECC

[2] algorithms, and RSA vs El-Gamal algorithms. In 2019 6th IEEE International Conference on Cyber Security and Cloud Computing

[3] (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom) (pp. 173-176). IEEE.

[4] Rachmawati, D., Budiman, M. A., & Saffiera, C. A. (2021). An Implementation Of Elias Delta Code And ElGamal Algorithm In Image Compression And Security. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012040). IOP Publishing.

[5] Mustafa, A. E., ElGamal, A. M. F., ElAlmi, M. E., & Bd, A. (2021). A proposed algorithm for steganography in digital image based on least significant bit. Research Journal Specific Education Faculty of Specific Education, Mansoura University, 21.

[6] Kurnia, H. Dafitri, and A. P. U. Siahaan, "RSA 32-bit Implementation Technique,"
Int. J. Recent Trends Eng. Res., vol. 3, no. 7, pp. 279–284, 2017

[7] Y.Kumar, R. Munjal, and H. Sharma, "Comparison of Symmetric and Asymmetric
Cryptography with Existing Vulnerabilities and Countermeasures," Int. J. Comput. Sci.Manag. Stud., vol. 11, no. 3, pp. 60–63, 2011.

[8] Ye, G., Wu, H., Liu, M., & Shi, Y. (2022). Image encryption scheme based on blind signature and an improved Lorenz system. Expert Systems with Applications, 205, 117709. https://doi.org/10.1016/j.eswa.2022.117709

[9] Adeniyi, A. E., Abiodun, K. M., Awotunde, J. B., Olagunju, M., Ojo, O. S., & Edet, N. P. (2023). Implementation of a block cipher algorithm for medical information security on cloud environment: Using modified advanced encryption standard approach. Multimedia Tools and Applications, 1-15. https://doi.org/10.1007/s11042-023-14338-9