WANNAHIDE

Fouad Mostafa[1], Abdullah Riad[2], Mahmoud Mohamed[3], Abdelrahman Fathy[4]

[1]Misr University for Science and Technology

89638@must.edu.eg

[2]Misr University for Science and Technology

89683@must.edu.eg

[3]Misr University for Science and Technology

89401@must.edu.eg

[4]Misr University for Science and Technology

89393@must.edu.eg

ABSTRACT

Ensuring secure communication has become paramount in today's digital landscape, where privacy breaches and data leaks are prevalent. This abstract presents Wanna Hide, a cutting-edge chat application designed to provide users with a secure and private messaging experience. The app incorporates the ElGamal encryption algorithm and the Double Ratchet protocol to offer end-to-end encryption and protect user data from unauthorized access.

The objectives of Wanna Hide are twofold: to guarantee the confidentiality and integrity of user messages and to ensure the resilience of the encryption mechanisms against potential attacks. To achieve these goals, the ElGamal encryption algorithm is employed, providing strong cryptographic security and safeguarding user messages from eavesdroppers. Additionally, the Double Ratchet protocol is implemented to establish forward secrecy, preventing the compromise of past conversations even if future encryption keys are compromised.

Through a meticulous implementation and rigorous testing, Wanna Hide has demonstrated remarkable findings. The app successfully provides end-to-end encryption, ensuring that only intended recipients can decipher the encrypted messages. The ElGamal algorithm, known for its computational security and resistance to attacks, offers a robust foundation for secure communication. Moreover, the Double Ratchet protocol, with its dynamic key exchange and continuous encryption key updates, enhances the resilience of the chat app against potential threats.

By leveraging these advanced cryptographic techniques, WannaHide wont only enables users to exchange messages securely but also provides a trustworthy platform for sensitive and confidential discussions. The app’s commitment to user privacy and data protection establishes a solid foundation for secure communication in an era where digital privacy is of utmost importance.

KEYWORDS: End-to-end encryption; Double ratchet; Elgamal Algorithm; AES-256-CBC; Hybrid encryption

1- INTRODUCTION

In today's digital landscape, safeguarding our communication privacy is a pressing concern. Chat applications with robust security measures and end-to-end encryption are crucial to protect user data. This introduction presents Wanna Hide, an innovative chat app that employs the ElGamal encryption algorithm and the Double Ratchet protocol to ensure privacy.

Recent studies emphasize the significance of end-to-end encryption in secure messaging apps. While various encryption algorithms and protocols exist, Wanna Hide distinguishes itself through its use of ElGamal encryption and the Double Ratchet protocol, setting it apart from prior works.

Wanna Hide relies on the ElGamal encryption algorithm, renowned for its strong cryptographic security. By employing ElGamal, the app encrypts user messages using a secure public key system, rendering them indecipherable to unauthorized parties. This approach ensures both message confidentiality and integrity.

Furthermore, Wanna Hide implements the Double Ratchet protocol to reinforce its security. The protocol enables forward secrecy by generating unique encryption keys for each message exchange. Even if an encryption key is compromised, previous messages remain secure due to the continuous key updates provided by the protocol.

Combining the robust ElGamal encryption algorithm with the forward secrecy of the Double Ratchet protocol, Wanna Hide guarantees a high level of privacy and security. Through a thorough examination of existing literature, Wanna Hide emerges as an innovative chat app that addresses previous limitations, offering a distinctive combination of encryption algorithms and protocols.

2- METHEDOLOGY

ElGamal Algorithm: The ElGamal encryption algorithm is a public-key cryptosystem 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 $D = B^a \bmod p$.

2. Compute $M = CD^{-1} \bmod p$.

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.

Double Ratchet: End-to-end encryption is a cryptographic technique used to secure communication between two parties, such as in messaging applications. One of the most commonly used techniques is double ratchet model to provide forward secrecy and message integrity.

In this protocol, first step which is key generation is similar to the first step in El-gamal encryption process but in addition both 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.

3- RESULT

Point of comparison	Elgamal double ratchet	Diffie-Helman double ratchet
Prime number length (bits)	At least 2048	At least 2048
Security base	Cylic group	Finite group
Functionality	Encryption Decryption	Key exchange
Key generation	Public key sharing	Key pairs generated in both parties

Table1. Comparison between ELgamal and Diffie-Helman Algorithms

4- DISCUSSION

- Security base for the Elgamal and Diffie hellman algorithms involves public-private key but they differ in the mathematical concepts used in both, that is cylcic group for Elgamal which basically mean every element in that group can be expressed as a power of the generator, while the Diffie-Hellman is based on infinite field that has unbounded number of elements .

- Functionality : main purpose of Elgamal crypto system is encryption decryption of a specific plain text while the diffie hellman main purpose is for key exchange the shared key that is used in double ratchet

- Key generation : public key sharing between parties is the main concept of Elgamal algorithm while keeping the private key to decrypt received messages, on the other hand Diffie-Hellman key exchange requires both parties to create their own key pair (private and public) in order to send the shared key to begin encryption and decryption process

- Speed : The longer the key the slower the public key operations and key generations. For the majority of users, the main factor in the selection of public-key size is the security, based on best attacks known. Speed is very rarely a determining factor.

- Another important factor in this system is the need for a good random number generator, since it utilizes randomization in the encryption process and it is critical to have different independent random numbers in the encryption algorithm.

5- CONCLUSION

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

ACKNOWLEDGEMENT

We extend our sincere gratitude to Prof. Dr. Rania Elgohary for granting us the exceptional opportunity to work on this remarkable project . Throughout this journey, we have acquired invaluable knowledge and skills that will greatly benefit our professional growth. We are profoundly grateful to Prof. Rania Elgohary for her unwavering guidance, continuous supervision, and provision of essential project-related information. Moreover, we deeply appreciate her support during the entire project completion process.

Prof. Dr. Rania Elgohary's constant guidance and willingness to share her vast knowledge have allowed us to comprehend the project and its intricacies in great depth, allowing us to successfully accomplish our assigned tasks within the designated time frame. We also express our thanks and appreciation to our colleagues who actively contributed to the development of this project, as well as those individuals who willingly offered their assistance and expertise.

Once again, we extend our heartfelt thanks to Prof. Dr. Rania Elgohary for her invaluable support and the opportunity to be part of this rewarding project.

6- REFERENCE

[1] https://www.ibm.com/docs/en/zos/2.1.0?topic=03353xxx-03353034

[2] https://www.ibm.com/docs/fi/zos/2.2.0?topic=openpgp-supported-key-sizes

[3] P. K. Panda, S. Chattopadhyay, A hybrid security algorithm for RSA cryptosystem, in Proc. ICACCS, (2017) 1-6.

[4] B. Schneier, Applied cryptography: protocols, algorithms, and source code in C, John Wiley & Sons, 2017.

[5] Y. Murakami, M. Kasahara, Hybrid inter-organization cryptosystem using ElGamal cryptosystem, in Proc. IEEE-ICCE, (2015) 378-379