Web-based Live Chat Application uses Advanced Encryption Standard Methods and Rivest Shamir Adleman

Edwin Kho*, Viny Christanti Mawardi, Novario Jaya Perdana
Informatics Engineering Department, Faculty of Information Technology
Universitas Tarumanagara

* Edwin.5355150008@stu.untar.ac.id

Abstract
The development of technology, especially in the data security system in maintaining the security of information data has been growing rapidly. In maintaining the security of information data there are branches of science in its development such as cryptography and steganography. In its application is not only done on one security technique, but can be done in combination with information security data. This study aims to create a data security system by implementing cryptography on messages in the live chat application. The message security in live chat applies a combination of AES and RSA algorithms. The AES (Advanced Encryption Standard) method is a cryptographic algorithm that can be used to secure data, the AES algorithm is a symmetric chipertext block that can encrypt and decrypt information, and the RSA Method is the RSA Algorithm is a block cipher algorithm (an algorithm that works per block of data) that groups plaintext into blocks first before being encrypted to become chipertext. This application aims to secure messages sent by the sender to stay safe when received by the recipient. The system will be based on a website using the PHP programming language, using a codeigniter framework and a MySQL database. This information system is in the form of live chat which will be encrypted by 2 methods. The sender will send a message through the live chat website then the message is encrypted so that its security is guaranteed, the message stored in the database will be automatically encrypted by the two methods applied then after arriving at the recipient the message is automatically described. The message sent is in the form of text and png files.

1. Introduction
Live chat is one of the most effective and efficient ways of communicating today. The use of chat applications has been very massive, almost everyone uses at least one application to exchange messages. This System is success to build encryption that based on android Alvianto, Andi Riski, and Darmaji using encryption in order to make messaging app[1]. The most widely used public key cryptography is RSA. The degree of difficulty to attack RSA encryption depends on the difficulty of finding the prime number factor used for this encryption. RSA uses two prime integers to get the public key and private key to be used in the message encryption and decryption process. RSA is used in this application to encrypt secret messages in the form of files so that the security of the secret message was stronger[2]. Cryptography is very desirable, and data security becomes incomplete without accompanying cryptographic methods. The cryptographic method consists of two methods; one is symmetric, and another is an asymmetric process. In the symmetric key cryptography, requires the same key to encrypt and
decrypt a message[3]. AES (acronym of Advanced Encryption Standard) is a Symmetric encrypt algorithm. AES bits for encrypt/decrypt the data and fortifies lengths are 128,192 and 256 bits[4]. RSA has been widely used as a security algorithm in the design of software and operating systems. In addition, RSA is also used for safe hardware design [5]. This aim is to Build a live chat system that is safe for users and a communication in real time.

2. Methods

2.1. Advanced Encryption Standard

Advanced Encryption Standard (AES) was published by NIST (National Institute of Standards and Technology) in 2001. According to research on the Design of Data Security Using AES Algorithm, it is understood that AES is a cryptographic algorithm that can be used to secure data. The AES algorithm is a symmetric chipertext block that can encrypt and decrypt information. Encryption changes data that can no longer be read called ciphertext, on the contrary decryption is changing ciphertext data into its original form known as plaintext.

The process carried out in each round is identical (from round 1 to round to Nr-1) except for round Nr. The identical process consists of SubBytes, ShiftRows, MixColumns and AddRoundKey. The identical process consists of SubBytes, ShiftRows, MixColumns and AddRoundKey. The encryption process is done using the AES algorithm, namely:

1. **AddRoundKey**
   
   This process is carried out at the beginning of the round by performing XOR operations per byte in the state matrix (plaintext) with each byte in the cipherkey, this stage is also called the initial round.

2. **Subbytes**

   SubBytes is a byte transformation where each element in the state will be mapped using a substitution table (S-Box). For each byte in the state expressed by S '[r, c]. S '[r, c] is an element in the substitution table which is the intersection of rows (x) with columns (y).

3. **Shiftrows**

   Shiftrows are basically the process of shifting bits where the leftmost bit will be moved to the rightmost bit (bit rotation). But the number of shifts is different, depending on each line. The first line is not shifting. Every byte of the second row in the state matrix is shifted one byte to the left. Then the third line is shifted to the left by two bytes and the fourth line is shifted to the left by three bytes. This process aims to produce diffusion by spreading the effect of nonlinear transformations on the rows of the state matrix for the next round.

4. **Mix Columns**

   In the MixColumns process, each column of the state matrix is performed a multiplication operation. It aims to spread the influence of each bit of plaintext and cipherkey to the ciphertext generated, in the direction of the state matrix column. Each state matrix column is treated as a four-term polynomial in the Galois field, then multiplied by modulo (X8 + X4 + X3 + X + 1). The MixColumns operation can also be viewed as matrix multiplication, by multiplying four numbers in the Galois field.
MixColumns is also called the process of multiplying each column with the following matrix:

5. *AddRoundKey*

In this AddRoundKey stage, the cipherkey that has been expanded first will get a roundkey that will be used for further processing. Then every byte of the state matrix output of the MixColumns process is performed XOR operations with each byte of the roundkey. Round or SubBytes, ShiftRows, MixColumns, and AddRoundKey processes are carried out until the nth round in the same way. While for the last round or also called the final round the SubBytes, ShiftRows, and AddRoundKey processes are still being done but the MixColumns process is not done.

The decryption process is performed using the Advanced Encryption Standard (AES) algorithm, namely:

1. **Add a Round Key**

The inverse or inverse of the AddRoundKey stage is an XOR operation between state matrix bytes arranged from ciphertext with roundkey bytes which was raised before. Roundkey used in each iteration contrary to the roundkey in the encryption process. Inverse of This transformation is used for the decryption process.

2. **Inverse Sub Bytes**

Inverse SubBytes is also a transformation bytes opposite to SubBytes transformation. In InvSubBytes, each element in the state is mapped by using the Inverse S-Box table.

3. **Inverse Shift Row**

For the decryption process, the Inverse process of the ShiftRows transformation is performed. InverseShiftRow is the transformation of bytes opposite to Transform ShiftRows. In the InverseShiftRows transformation, it is done bit shifts to the right whereas ShiftRows shifts bits to the right left.

4. **Inverse Mix Columns**

To decrypt a message, an inverse of the transformation is performed MixColumns is multiplying each column of the results from the Inverse AddRoundKey with the following matrix:

2.2. *Rivest Shamir Alderman*

RSA is an algorithm for public-key encryption. This algorithm is the first algorithm known to be the most suitable for signing (signing) and for encryption (encryption) and one of the first major discoveries in public key cryptography.

RSA is still widely used in electronic commerce protocols, and is believed to be very safe because it is given long keys and its applications are very up-to-date.

First of all, plaintext is encrypted into blocks, where each block has a binary number less than \( n \) for \( n \) a value. That way the block size must be less than or equal to \( \log_2 (n) \). Encryption and decryption of a plaintext block \( M \) and ciphertext block \( C \):

1. \( C = \text{Mod } n \)
2. \( M = C^d \text{mod } n = (M^e)^d \text{mod } n = Med \text{mod } n \)
To get the above, the conditions that must be met are as follows: 1. The values e, d, n can be found, so that Med can get M = n mod for every M < n. 2. Relatively easier to calculate Me and Cd for each value of M < n. 3. It is difficult in practice to find d given e and n. ed = k (n) + 1.

The steps of the RSA algorithm are as follows:
1. Key Generator:
   - Two large prime numbers are chosen at random, p and q. The value of p and q must be kept a secret.
   - Calculated n = p x q. The amount n does not need to be kept a secret.
   - Calculated m = (p - 1) (q - 1). The magnitude of m needs to be kept a secret.
   - An integer is chosen as the public key, called its name e, which is prime relative to m. e is prime relative to m, which means the dividing factor the second largest is 1, mathematically called gcd (e, m) = 1. For look for it can be used Euclid's algorithm. The value of e is not confidential.
   - The private key is calculated, called by name d such that (dxe) mod m = 1. To find the appropriate d value, you can also use the Extended Euclid algorithm. D value is confidential.
2. Encryption:
   a. Take the message recipient's public key, e, and n
   b. Express plaintext M into M1, M2, ... blocks so that each blocks represent values in the interval [0, n - 1].
   c. Each Mi block is encrypted into a Ci block with the formula Ci = Mi ^ e mod n
3. Decryption:
   a. Each Ci ciphertext block is decrypted back to Mi block with the formula Mi = Ci ^ d mod n ^ 5

3. Results and Discussion

3.1. Result Advance Encryption Standards, Rivest Shamir Adleman
For the trial of the Advance Encryption Standard and Rivest Shamir Adleman methods enter plain text and key first. Edwin Kho 45 64 77 69 6e 20 4B 68 6F Key: rahasia 72 61 68 61 73 69 61

| Encryption Result | C = P^0 mod n |
|-------------------|---------------|
| 40,96,16,8,16,80,96,0,40,16,96,112,32,88,27,40 |

| RSA Decryption | P = C^d mod n -> the result of decryption |
|----------------|----------------------------------------|
| 140,197,54,189,186,23,128,30,194,173,128,53,227,212,3,149 |

From the existing decimal data is converted to hexa and will produce:
AES Decryption

Cipher Text:

```
409616816809604016961123232882740
```

Changed to Hexa produces the following:

```
8CC536BDBA17801EC2AD8035E3D40395
```

Produce:

```
EBA3D33BC2F04560507E65181ABFFADC
```

InvMixColumns each column in the state is multiplied by the multiplication matrix in AES

Produce:

```
31C36FCA87F08EA84FEDA34FAF2FE816
```

The process was repeated 10 Rounds and compressed the decryption results back to the original message with the following decryption results Get Decryption Results:

```
456477696E204B686F EdwinKho
```

3.2. Result

Results after testing the system that has been made using the AES and RSA methods for security of Live Chat Messages. the system that has been made will be run first to ensure every function contained in the program can be run properly.

Try it out on the system by trying to enter the program to process the sending of messages and see whether the encryption and decryption process is running or not - the steps can be seen below:

1. Open the Live Chat Website application
2. Register for the application
3. After registering the user will enter on this dashboard is the main page the first time the user logs in
4. After successfully entering the system the user must first add contacts to send messages to the intended person

![Figure 3. Adding Contacts](image3.png)

5. After successfully adding a friend the user can immediately try pressing the chat button and enter the inbox and send the message as shown below

![Figure 4. Inbox page](image4.png)

6. After successfully sending the message the message will be sent to the recipient and the message will be automatically encrypted and the results can be seen in the following image below

![Figure 5. Message Encryption](image5.png)

Based on these results, the live chat application is successfully run and can also send messages to the recipient and the message is encrypted using both methods and decrypted again so that the message can be received by the recipient and with a security system.

4. Conclusion
The conclusions that can be obtained based on the testing of the AES and RSA methods have been successfully implemented to encrypt messages and decrypt messages into a live chat system. And the sender succeeds in sending a message to the recipient whose message has been automatically encrypted into the system, the system only encrypts messages in the form of text and jpg files.

5. References
[1] Alvianto, Andi Riski, Darmaji, "Safeguarding SMS Messaging with Android-Based RSA Algorithm." Journal of Science and Art ITS 4.1 (2015): A1-A6.
[2] Arya Reza Nugraha, Ary Mazharuddin S, Concealment of Encrypted Secret Messages Using the RSA Algorithm in Compression Media. Surabaya. 2013.
[3] Bansal, V. P., & Singh, S. (2015, December). A hybrid data encryption technique using rsa and blowfish for cloud Computing on fpgas. In Recent Advances in Engineering % Computational Sciences (RAECS), 2015 2nd International Conference on(pp. 1-5). IEEE

[4] Anup Ashok Patil, Shital Mali, “Hybrid Cryptography Mechanism for securing self-Organized Wireless Networks”, IEEE, 2016.

[5] Zainal Arifin. Case Study of Using the RSA Algorithm as a Safe Cryptographic Algorithm. Samarinda. 2009