Cryptography Application to Message Text using the Android-Based RC4 Method

Rizky Suhardianto¹, Jonson Manurung²
¹,²Informatics Engineering Study Program, STMIK Pelita Nusantara, Medan, Indonesia

ABSTRACT
A text message or chat is an action or activity in the form of chat, but what is different is that the chat uses the latest application, be it a gadget or other media. Etymologically, the word "chat" comes from English which in Indonesian means chat, converse, chat, or kongko. But security is still not sufficient, where other people can see the original message of the original text of the chat, by applying cryptography to chat messages, the chat will be safe from other parties. Android is also designed to make it easier for developers to create applications with minimal restrictions so that developer creativity can be more developed. Therefore we need security that can secure messages or chat in order to help someone's privacy in carrying out daily activities. Cryptography is an art of encryption to keep messages from being easily read or so-called secret messages. There are two main concepts, namely encryption, and decryption. Encryption is the process by which the information sent is converted into a form that is not recognized as the initial information by using a certain algorithm. Decryption is the opposite of encryption in that it converts the disguised form back into the original information. In cryptography, there are several methods following its use, one of which is RC4. Rivest Code 4 (RC4) is a type of code stream which means that the encryption operation is performed per 1 byte character for one operation. Using the Rivest Code 4 (RC4) method is the best way to secure data by changing some characters which will be difficult to hack later by those who want to commit fraud.

Keywords: Cryptography, Text Messaging, RC4.

1. INTRODUCTION
The development of science and technology has influenced all aspects of human life. Information and data can be easily and quickly sent to consumers via the network. This of course poses a risk if the information and data sent can be accessed by unauthorized parties, resulting in data leakage. Security issues are one of the most important aspects of an information system. Meanwhile, security issues, for now, are often ignored, even eliminated. So now it is necessary to emphasize data security in order to reduce the risk of data leakage.

A text message or chat is an action or activity in the form of chat, but what is different is that the chat uses the latest application, be it a gadget or other media. Etymologically, the word "chat" comes from English which in Indonesian means chat, converse, chat, or kongko. But security is still
not sufficient, where other people can see the original message of the original text of the chat, with the application of cryptography into chat messages, the chat will be safe from other parties.

Android is an operating system for mobile devices that includes middleware (virtual machine) and a number of major applications. Android is a modification of the Linux kernel. The purpose of making this operating system is to provide an open platform, for example in the use of chat, but in its use, there are system weaknesses. Android is also designed to make it easier for developers to create applications with minimal restrictions so that developer creativity can be more developed. Therefore we need security that can secure messages or chat in order to help someone's privacy in carrying out daily activities.

2. RESEARCH METHOD

Cryptography is the art of encryption to keep messages from being easily read or so-called secret messages[1]. There are two main concepts, namely encryption, and decryption. Encryption is the process by which the information sent is converted into a form that is not recognized as the initial information by using a certain algorithm[2], [3]. Decryption is the opposite of encryption in that it converts the disguised form back into the original information. In cryptography, there are several methods following its use, one of which is RC4[4], [5]. Rivest Code 4 (RC4) is a type of code stream which means that the encryption operation is performed per 1-byte character for one operation. Using the Rivest Code 4 (RC4) method is the best way to secure data by changing some characters which will be difficult to hack later by those who want to commit fraud[6], [7].

Algorithms are the flow of thoughts in completing a job, which is outlined in a written form that can be understood by others[8]. The RC4 algorithm is an asymmetric cryptographic algorithm, Rivest Code 4 (RC4) cryptographic algorithm. It is called an asymmetric cryptographic algorithm because it uses the same key to encrypt or decrypt a message, data, or information. RC4 uses a key length from 1 to 256 bytes which are used to initialize a 256-byte table or commonly referred to as a state-array[9][10][11][12]. The state-array will be randomized again and processed to produce an encryption key which will be XORed with plaintext or ciphertext. Broadly speaking, the RC4 algorithm is divided into two, namely Key Setup or Key Scheduling Algorithm (KSA) and Stream Generation or Pseudo-Random Generation Algorithm (PRGA) XOR process with stream data.

The following is the algorithm process using the RC4 method[13], [14]

1. Key Setup / Key Scheduling Algorithm (KSA)

In this section there are three stages of the process, namely:

a. Initialize the S-Box

In this stage, the S-Box will be filled with values according to the index to get the S-Box. The algorithm is as follows:
1. for i = 0 to i = 255 do,
2. fill in s with the value of i,
3. add i to 1, go back to step

![Figure 1](image)

Figure 1. An example of S-Box initialization by entering sequence data for each block

b. Stores the keys in the Key Byte Array

In this stage, the key that will be used to encrypt/decipher will be entered into an array of size 256 repeatedly until the entire array is filled. The algorithm is as follows:
1. fill j with 1,
2. for i = 0 to i = 255 do,
3. if \( j > \) key length then,
4. \( j \) is filled with the value 1,
5. end if,
6. Fill in \( k \) to \( i \) with ascii value of the key character to \( j \),
7. the value of \( j \) is increased by 1,
8. add \( i \) to 1, back to 2.

The algorithm will obtain a sequence of array keys, for example, the following for keys with a length of 8 characters with a sequence of characters in ASCII "109 97 104 98 98 97 104".

![Figure 2. Example of key storage results in a byte array key](image)

c. Permutations on S-Box

At this stage, a value will be generated which will be used as a rule for permutations on the S-Box. The algorithm is as follows:
1. fill in the value of \( j \) with 0,
2. for \( i = 0 \) to \( i = 255 \) do,
3. Fill in the value of \( j \) with the result of the operation \( (j + s(i) + k(i)) \mod 256 \),
4. exchange rates of \( s(i) \) and \( s(j) \),
5. add \( i \) to 1, back to 2.

From this algorithm, the S-Box value will be obtained which has undergone a transposition process so that the order is randomized, for example for the key in the example as follows.

![Figure 3. An example of S-Box randomization based on the key array used](image)

2. Stream Generation / Pseudo-Random Generation Algorithm (PRGA)

This stage will generate a pseudo-random value that is subject to XOR operation to produce ciphertext or vice versa to produce plaintext. The algorithm is as follows:

- a. fill in index \( i \) and \( j \) with value 0,
- b. for \( i = 0 \) to \( i = \) length of plaintext,
- c. Fill in the value of \( i \) with the results of the operation \( (i + 1) \mod 256 \),
- d. Fill in the value of \( j \) with the result of the operation \( (j + s(i)) \mod 256 \),
- e. exchange rates of \( s(i) \) and \( s(j) \),
- f. Fill in the value of \( t \) with the result of the operation \( (s(i) + (s(j) \mod 256)) \mod 256 \),
- g. Fill in the value of \( y \) with the value of \( s(t) \),
- h. \( y \) value is subject to XOR operation on the plaintext,
- i. add \( i \) to 1, back to 2.
Thus, the resulting ciphertext with XOR results between stream keys from S-Box and plaintext sequentially.

3. Encryption And Decryption Process With XOR Operation
   The encryption/decryption process begins by converting each plaintext to binary. The formula for encryption and decryption is as follows:
   1. Encryption process formula
      \[ C = P \cdot K \] (Ciphertext = Plaintext xor key)
   2. Decryption process formula
      \[ P = C \cdot K \] (Plaintext = Chip text xor key)

3. RESULTS AND DISCUSSION
   Here is a manual calculation of the RC4 algorithm in 256 bytes mode. First initialize the S-Box with a length of 256 bytes, with \( S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, \ldots, S[255] = 256 \), so that the array \( S \) becomes: 0 1 2 3 ... 255.
   The first step that must be done is to initialize the Key so that the \( K \) state-array is formed as follows:
   Initialize the S-box
   0 1 2 3 4 5 ... 255
   After initialization, KSA is then performed to create a random state-array. But before doing random calculations, first, convert the keys into ASCII numeric form. Convert it to a decimal number for the easy permutation.
   The next calculation randomizes the key with the previously arranged array from 0-255. The pseudocode is used to perform the above permutations. Here's the pseudocode:
   
   ```
   i = 0
   for i from 0 to 255
       j = (j + S[i] + K[i]) mod 256
       Swap (S[i], S[j])
   Endfor
   ```
   
   After performing the ASF, will be carried out PRGA. PRGA was conducted 256 times. This is because it takes 1 key and 1 XOR operation for each character in the Plaintext.
   After doing the permutation, the random key is taken here based on the length of the text that will be calculated, so this will get a key of the same length. The pseudocode used in performing the random permutation above.
   Stages of system implementation discuss the stages when running the system being built.
   1. Display Message Form
      The message display is the first display in order to send messages by typing plain text. The following is a message display.

![Figure 4. Message Display](image)
2. Display Encryption
   This display shows how to fill in the message correctly. There is also a view that can be seen in the image below:

   ![Crypto Message](image1)

   **Figure 5.** Encryption Display

3. Inbox view
   In this view, the inbox is sent to the inbox, the following is the inbox view that can be obtained as follows:

   ![Inbox](image2)

   **Figure 6.** Inbox

4. In-App Display Inbox
   In this view is the inbox in the application, the following is the inbox view that can be obtained is as follows:
5. Display Decryption
   In this view we select the message to be decrypted and here is the appearance of the form:

   ![Figure 7. Inbox in application](image)

6. Display Results
   In this view is a display of the results of the decryption, the following displays the results that can be obtained are as follows:

   ![Figure 8. Decryption](image)
4. CONCLUSION

The RC4 method is one type of synchronous stream cipher, which is a cipher that has a symmetric key and encrypts the plaintext digit by digit or bit by bit by combining binary operation (XOR) with a semi-random number to produce the same data. On the other hand, the ciphertext is returned to all text form by means of decryption so as to produce plaintext with its original shape. Creating a text messaging application with Android to convert text into coded text using the RC4 method is designed to use XML and JavaScript language coding.

REFERENCES

Cryptography Application to Message Text using the Android...

( Rizky Suhardianto)

Cryptography Application to Message Text using the Android...

Rizky Suhardia

197

