Transpose-Trigram Cipher

Trigram-based Substitution Cipher

Jonathan Alvaro
Informatics Engineering
ITB
Bandung, Indonesia
13516023@std.stei.itb.ac.id

Abner Adhiwijna
Informatics Engineering
ITB
Bandung, Indonesia
13516033@std.stei.itb.ac.id

Abstract—**Polyalphabetic substitution ciphers are still vulnerable to decryption methods such as frequency analysis. As such, the paper proposes a layered trigram-substitution cipher that increases the security while still allowing acceptable encryption speed.**

Keywords—trigram; block cipher; substitution; caesar cipher; transposition; rotation;

I. Introduction

In the history of cryptography, one of the most commonly method used to ensure the secrecy of a message is the substitution method. In this method, each letter in the original message is replaced by another letter. There have been various variations of this method of encryption, with some of them being more famous than the others. Notable examples include the Affine cipher and the Caesar cipher.

But, this method of encryption proved to be susceptible to various code-cracking methods such as frequency analysis. This is caused mainly by two weaknesses of this method, namely the unchanging position of the letters between the original message and the predictable frequency of each letter, which is unique for each language. As such, this paper proposes a new block cipher that makes use of a variation of the traditional substitution methods in order to improve the security of the cipher while still keeping the ease of implementation of the substitution method.

II. Substitution-Based Ciphers

This section will review other encryption techniques that are based on the substitution method, no matter whether it is a traditional or a modern encryption technique.

A. Caesar Cipher

The first cipher to be reviewed in this chapter is, arguably, one of the most well-known ciphers out there. The cipher receives its name from the first recorded user of the encryption method, Julius Caesar. This cipher was used by the Roman army in order to protect messages of military significance [1]. Despite how well-known the cipher is, it actually employs a very simple encryption method. In fact, any message encrypted with this method could be easily decoded in less than an hour with only pen and paper, as there are only 25 possible unique keys and how simple the encryption method is.

Basically, the Caesar cipher is a monoalphabetic substitution cipher. That is, each letter in the original message is replaced by another message in order to hide the content of the original message from any unwanted third party. In order to determine the encrypted text, each letter in the original message is replaced by a letter a certain distance in front of the letter to be encrypted within the alphabet. The distance in question in this case is called a key, namely an integer between 1 and 25, inclusive, which is predetermined and agreed upon by both the sender and the receiver. For example, with a key of 3, the letter a will be encrypted into the letter d. In the case that the end of the alphabet is reached, this method wraps around to the beginning of the alphabet. An example of this for a key of 3 is the letter y. In this case, it would be replaced by the letter b.

On the receiver's end, the original message could be known by decrypting the ciphertext. This is done by replacing each letter in the ciphertext by another letter a certain distance from it within the alphabet. The only difference between the encryption and decryption process lies within the fact that the replacement letter is not located in front of the letter to be decrypted, but behind the letter in question. For a key of 3, a letter c within the ciphertext would be replaced by the letter 3 positions behind it within the alphabet, namely z. Aside from this difference, the encryption and decryption process are identical.

The simplicity of this method allows messages to be encrypted and decrypted rapidly by manual calculations, which makes it suitable for the past, where knowledge of cryptography is limited and complex calculating machines did not exist.

B. Vigenère Cipher

The Vigenère cipher is another traditional encryption method that was widely used before computers and modern cryptography techniques become widespread. As with the Caesar cipher, the letters in the original message would be replaced by another letter in the encrypted message. The difference here is that the Vigenère cipher is a polyalphabetic cipher instead of a monoalphabetic one. What this means is that for each letter in the alphabet, there are more than one letter that could become its replacement within the encrypted version of the message. This method of encryption is far more secure compared to the Caesar cipher, because it is resistant to frequency analysis. The reason for this is that each letter could be replaced by more than one letter and as such, the frequency of each letter within the encrypted message does not necessarily reflect the frequency of each letter within the original message. This resistance to frequency analysis prevents this encryption method to be unbreakable for a little bit over three centuries, from 1553, the year it was first published, to 1863, when the Kasiski method was finally released to the public [2][3]. This method is also more resistant to brute-force decryption compared to the Caesar cipher as there is essentially an unlimited number of possible key variations for this method.

In order to encrypt a message using this method, a certain helping tool called the Vigenère square [4]. This square is a 26 by 26 matrix that gives a replacement letter given the original letter and a letter within the key, which is a string of any length. Should the length of the key be shorter than the text to be encrypted, the key itself is repeated multiple times until it is of at least the same length of the plaintext. To encrypt a letter using the Vigenère square, there are three steps to follow:

1. Find the row which label is the original letter.
2. Find the column which label is the letter within the key with the same position of the original letter within the original string.
3. Replace the original letter with the letter at the intersection between the row and the column from steps 1 and 2.

On the other side, to decipher the encrypted message is also merely a matter of finding the matching letters within the Vigenère square. The steps to decipher the encrypted message are:

1. Find the column which label is the key for the letter being deciphered.
2. Within the column, find the cell that contains the letter being deciphered.
3. The label of the row containing the matching cell is the original letter.

With this method, the encrypted message is now resistant to traditional frequency analysis because different occurrences of a letter would use different keys to encrypt it and thus would be replaced by different letters. But, there is still a way to decrypt a message encrypted in this manner, namely by using the Kasiski method. The Kasiski method is basically a modified version of the frequency analysis that makes use of the repeating nature of the key used to encrypt the text. Because of this reason, the shorter the key is compared to the original text, the less secure the encrypted message will be.

III. Transpose-Trigram Cipher

Both of the encryption methods reviewed in the previous chapter has the advantage of being easy to both encrypt and decrypt, thus making it useable for communications where messages are transmitted rapidly between both parties. But, in this day where the computing power of computers are widely available, those methods of encryption are very insecure and cannot be relied on to deliver any message of importance. Because of that, this paper proposes the Transpose-Trigram cipher which takes advantage of the ease at which those ciphers are done, but provides increased security to techniques such as the Kasiski method and the frequency analysis.

A. Cipher Overview

The Transpose-Trigram cipher is a block cipher which uses the Vigenère cipher, Caesar cipher, transposition, and a trigram-based substitution as the encryption and decryption function within a Feistel cipher. In addition to this, transposition and rotation of the message itself is also done multiple times. This is done in order to remove the weakness of the Vigenère cipher, namely repeated occurrences of a certain letter being encrypted using the same letter within a key. The repeated trigram-based substitution also helps protect the encrypted message from brute-force decryption by the large number of possible substitution tables for each encryption.

Further adding to the security of the cipher is its ability to be run in five different block cipher mode of operations, namely ECB, CBC, CFB, OFB, and counter mode.

B. Cipher Details

The Transpose-Trigram cipher is a block cipher with a block size of 12 bytes. This block size is chosen due to the trigram-based substitution requiring the block size to be divisible by 3. This cipher also requires a 12-bytes long key that would be used to generate the trigram-substitution tables and to initialize the key for both the Caesar cipher and the Vigenère cipher.

For the encryption process, each block of the message would be processed through 12 iterations of the Feistel cipher. This number of iterations is an arbitrarily determined number that according to our simulations, is secure enough while still maintaining the ease of encryption and decryption of the ciphers within the previous chapter. The following are steps on how a block of message is processed through the Feistel cipher:

1. Divide the block into two halves of equal length.
2. Use the following formula to encrypt each half,

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

$$i = current\ iteration, 0\ being\ the\ initial\ value$$
$$L_i = The\ left\ half\ of\ the\ block\ on\ iteration\ i$$
$$R_i = The\ right\ half\ of\ the\ block\ on\ iteration\ i$$
$$K_i = The\ key\ on\ iteration\ i$$
$$F = The\ encryption\ function$$

After going through 12 iterations of this encryption process, both halves are joined together in order to create the final encrypted block.

The encrypted message can be decrypted back simply by doing the same process but with a slightly different formula for the Feistel cipher, namely:

$$L_i = R_{i+1} \oplus L_{i+1}$$
$$R_i = F^{-1}(L_{i+1}, K_{i+1})$$
$$F^{-1} = Decryption\ function$$

As in the encryption process, in the decryption process, the Feistel cipher is done for 12 iterations in order to transform the encrypted message back into the original plaintext.

C. *Encryption Function*

The encryption function used within this cipher is made up of two different phases. The first phase is the simpler one which uses both the Caesar cipher and letter transposition in order to scramble the letters within the original message. As for the second phase, which is the more complex of the two, it uses both the Vigenère cipher and a trigram-based substitution in order to further secure the message by increasing the difficulty for frequency-based analysis such as the traditional frequency analysis and the Kasiski method. Each iteration of the Feistel cipher described in the previous section uses one of the two phases, with odd-numbered iterations using the first phase and the even-numbered iterations using the second phase. This means that, effectively, there are 6 different iterations that makes use of both phases. But the reason that the phases are used in turns instead of a single is to introduce the further obfuscation that the Feistel cipher provides between each phase.

First, for the first phase, the block that is entered into the encryption function is transposed by converting it into a matrix with two columns. Then, the new transposed string is obtained by reading the matrix right to left, top to bottom. Then, the resulting string will be encrypted using the Caesar cipher in order to scramble the letters within the block half being processed.

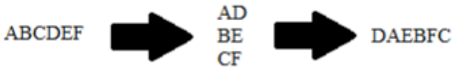

Fig. 1. Transposition of string within the first phase of the encryption function

As for the second phase, the first step is to further divide the block half being processed into two halves of 3 bytes each. In order to help with this description, these 3 bytes chunks will be called sub-halves from hereon. The reason for this is to introduce interdependency between bits in the encrypted message. This is done in order to prevent bit-changing attacks to succeed by causing the decryption process to fail if even a single bit is corrupted. To begin with, the first sub-half will be encrypted using the extended Vigenère cipher with the second sub-half as a key. Then, the result of the extended Vigenère cipher will be protected from the Kasiski method by substituting the resulting trigram with another trigram by using a lookup table. Then, the second sub-half will be encrypted as well by the same process using the encrypted first sub-half as the key for the extended Vigenère cipher. After that, both sub-halves are joined together to form the encrypted version of the original block. As for the lookup table for the trigram substitution, the encryption function will use one of the 6 tables pre-generated with the seed entered by the user on the beginning of the encryption process. As for the order which the tables are used, the first occurrence of the second phase within the encryption of a single block uses the first table, the second occurrence uses the second table, and so on. In other words, every n-th occurrence of the second phase of every block will use the same lookup table, namely the n-th generated table. This trigram substitution step helps protect the encrypted message from brute-force attacks as there are $2.74 * 10^{14}$ possibilities for each table. Considering the fact that the cipher uses 6 different tables for the substitution step, there are $4.32 * 10^{86}$ possible table combinations which makes a brute-force attacks of this cipher very difficult. This at the very least provides the necessary security for exchanging important messages at the level of personal communication between individuals.

IV. Implementation

In this section, the paper will discuss the results of the application of the cipher using several test-cases and modes of operations. All tests were done using scripts written in Python. Considering the fact that Python is an interpreted language, the results within the paper should not be taken as the maximum performance of the cipher as more efficient implementations might be possible. As a detailed explanation of how the cipher works has been provided in the third section of the preceding chapter, this chapter will assume that the readers have already understood the big picture of how the cipher works and immediately jump to the results obtained from the authors' implementation of the cipher.

A. *Methods*

In this chapter and all of its sections, all of the data here will be the best obtained result from running the cipher against a certain test-case for 3 times for encryption and one time for decryption. As for the specifications of the testing machine, it is as follows:

OS	Windows 10 Pro 64-bit (build 17134)
CPU	Intel Core i7-8700K
RAM	16GB

Fig. 2. Specifications of the machine used to acquire performance data

B. *Speed*

In this section, the paper will present the results obtained from the program and try to explain the reason for the obtained results. More specifically, it will test the claim made in the beginning of the paper that the proposed cipher maintains the ease of encryption and decryption of its component ciphers. The following table shows the time taken purely for the encryption process for text of various length and mode of operations.

Input	Encryption Time	Decryption Time
Length: 12000, OFB	0.16	0.19
Length: 12000, ECB	0.16	0.16
Length: 12000, CBC	0.17	0.16

Fig. 3. Performance of the Transpose-Trigram Cipher for various inputs

All time in the table are in seconds and rounded to the nearest hundredth and is calculated only for the duration the cipher is encrypting/decrypting the message which doesn't include the time taken to generate the lookup tables needed for the trigram substitution. As for the input parameters, the length of each input is in characters. For all three of the tests listed in the table, the input is a string of randomly generated character 12000 characters long.

As can be seen, the time it takes to encrypt and decrypt a message 12000 characters long is very fast. This makes the proposed cipher usable for practical uses, such as securing messages exchanged by phone. But this is only if we do not consider the time needed to generate the set of tables that is needed to perform the trigram substitution step. During testing, the table generation takes the majority of the script's running time, with each execution of the table generating function taking at least 1 minute. This introduces a very significant overhead into the encryption, 50-60 times the encryption/decryption time in fact, which would prohibit the use of this cipher for real-life applications that requires rapid message exchange. Not only that, it also means that the encryption method is preferable to be used for extremely long messages. The reason for this is that as the length of the message increases, the more time is spent encrypting/decrypting the message itself instead of generating the tables for the process.

C. Security

Finally, this section will try to test the, in the authors' opinion, most important aspect of an encryption method, namely, how secure the method is. Due to the limited knowledge the authors have in this field, the security analysis done will be limited to confusion and diffusion analysis on two cases, a 1-bit change in the encrypted message, a 1-bit change in the key, and the feasibility of brute-force attacks and frequency analysis. Furthermore, the analysis will be done solely in the ECB mode to reduce the complexity of the analysis.

Based on the results received from the tests, the cipher method is found to be sufficient in terms of confusion and diffusion. The reason for this is that a 1-bit change in the encrypted message breaks the decrypted message for the characters that are in the same block of the changed bit. The change does not affect other blocks since the test is done on ECB mode. Furthermore, this cipher is very secure when it comes to bit changes in the key itself, as when this attack method is tested, the whole cipher breaks and outputs garbled text that bears practically no resemblance to the original message. In fact, due to the fact that the cipher uses extended ASCII instead of just the letters in the alphabet, the output received from decrypting a message using the wrong key outputs characters that should not even exist in a normal text, thus further ensuring the security of this cipher from this attack.

Next, for brute-force attacks, based on the analysis that were done, this method of attack is practically impossible. First, it would require the attacker to test out $4.32 * 10^{86}$ possible table combinations. In addition to that, the Caesar shifts that are done by the phase one of the encryption function, further increases the possible combination of keys needed to break the cipher by another order of magnitude. According to the authors knowledge of the computing power that is available to most people, this number of possible combinations makes breaking the cipher by brute-force impractical and simply not worth the resources needed to achieve this.

Finally, as for frequency analysis, the cipher is secure from this avenue of attack because of the trigram substitutions. As the substitutions are done in sets of three characters, it means that each letter has an almost equal chance to be used as a replacement for another character. This ensures that frequency analysis of the resulting encrypted message would not expose any significant information. This protection from frequency analysis is further increased in the case of readable texts. This is due to the mapping of merely a subset of the 128 characters ASCII into the 256 characters extended ASCII which causes each character's frequency to be distributed to at least 2 or more different characters' frequency.

Based on all the tests and analysis that were done, it is determined that as long as the key used to encrypt a message is kept secret, the cipher would be safe enough for everyday use for messages that are not too important, e.g. personal messages. On the other hand, due to the limited security analysis performed in this paper, it is not recommended to use this cipher for very important messages such as those on the national or corporate level until a more detailed and extensive analysis is done.

V. Conclusions

As proven by the tests on the previous chapter, the Transpose-Trigram cipher is secure enough to be used in daily life while still maintaining the capability of being rapidly calculated as its component ciphers. On the other hand, there are still weaknesses to be fixed and further improvements for the cipher that could be done in order to improve its performance.

First of all, the trigram substitution step within the second phase of the encryption function uses a lookup table which takes a relatively very long time in order to generate. There are several ways to fix this point. The first is simply to find another method of substitution which does not rely on substitution tables which would enable bypassing the table generation altogether. Another possible solution is to use a set of tables for as long as it takes to generate another set of tables. This way, there will be a thread that always runs in the background which continuously calculates the next set of tables based on the initial key given. Whenever the thread finishes calculating a set of tables, it would send it back to the main encryption program when it detects that the main program is not in the middle of encrypting a message. This way, the delay of generating the lookup tables would only be felt during the initial startup of the program. Of the two solutions proposed here, the more desired one is the first solution, due to a couple of reasons. The first reason is that the lookup table takes up a lot of space in memory (over 1 GB for trigram-based substitutions), which costs precious system resource and makes it extremely impractical to generate the lookup tables for substitutions of ngrams with length 4 and above (requires over 16 GB, which is the RAM capacity of the testing machine). The second reason is that should there be a powerful enough machine, the intervals between the generations of table sets gives potential attackers to simply try out every single possible table combination (very impractical, but still possible).

The next possible improvement that the authors want to point out is the possibility of randomizing the phases used within the encryption function. At the moment, the way that the encryption function is designed, only one half of the block would receive the stronger encryption of the second phase, while leaving the first phase somewhat vulnerable with only transpositions and letter shifting. On the other hand, pure randomization might mean that the security of the cipher as a whole to be unstable given the variable number of second phase encryptions that each block receives.

The authors also propose that a modification be made to somehow allow each block to be linked to each other in the encryption process. This improvement is proposed in order to close off the possibility changed bit attack. If possible and implemented, this would significantly increase the security of the cipher for use in messages that contains sensitive instructions.

Finally, the author also proposes a further, more detailed analysis of the security that the cipher provides to a message, preferably by experts in the field of cryptography that would have more extensive knowledge compared to the authors. Another thing that would need to be analyzed would be how the cipher performs for extremely long and extremely short messages as the tests performed in this paper are very limited in terms of variability.

Acknowledgment

The authors would like to express their gratitude for the support offered by Mr. Rinaldi Munir, the professor in charge of the cryptography class the authors are enrolled in during the time of the paper's writing. For without the knowledge and resources he introduced to the authors, the authors would not be able to design and propose the cipher within the paper. Then, the authors would like to also thank the authors' parents for supporting the authors throughout college as without their support the authors would never be able to go so far in their study. And finally, the authors would like to thank all the experts out there that have shared their knowledge on the internet for others to learn as without it, the authors would have trouble finding the needed data to back their claims and to design a cipher that could perform sufficient security to be implemented for practical use.

References

[1] "Cracking the Code — Central Intelligence Agency", Cia.gov, 2019. [Online]. Available: https://www.cia.gov/news-information/featured-story-archive/2007-featured-story-archive/cracking-the-code.html. [Accessed: 12- Mar- 2019].

[2] L. Smith, Cryptography. New York: Dover, 2013.

[3] F. Kasiski, Die Geheimschriften und die Dechiffrir Kunst. Berlin: E.S. Mittler und Sohn, 1863.

[4] https://www.topspysecrets.com/images/vigenere-cipher.jpg. 2019.

STATEMENT

With this, we declare that the paper that we write is written based on our own ideas. This paper contains no plagiarized ideas, nor is it a translation of an existing paper as far as the authors' knowledge goes.

Bandung, 13 March 2019

Signed,
Jonathan Alvaro, 13516023

Bandung, 13 March 2019

Signed,
Abner Adhiwijna, 13516033