An implementation of super-encryption using RC4A and MDTM cipher algorithms for securing PDF Files on android

M A Budiman, D Rachmawati and M R Parlindungan

Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Jl. Universitas No. 9-A, Kampus USU, Medan 20155, Indonesia

Email: mandrib@usu.ac.id, dian.rachmawati@usu.ac.id and mutiararizkyparlindungan2013@student.usu.ac.id

Abstract. MDTM is a classical symmetric cryptographic algorithm. As with other classical algorithms, the MDTM Cipher algorithm is easy to implement but it is less secure compared to modern symmetric algorithms. In order to make it more secure, a stream cipher RC4A is added and thus the cryptosystem becomes super encryption. In this process, plaintexts derived from PDFs are firstly encrypted with the MDTM Cipher algorithm and are encrypted once more with the RC4A algorithm. The test results show that the value of complexity is $\Theta(n^2)$ and the running time is linearly directly proportional to the length of plaintext characters and the keys entered.

1. Introduction

Nowadays, sensitive information may become very vulnerable to be known to others, as it can be captured and manipulated by unauthorized parties. In cryptography, there are some processes that can be done to ensure the security of the information transmitted; one of them is by encoding the information into data that can not be read or understood by other parties, but can be read by the sender and the recipient. PDF files are considered more secure than the other file types since they can be set to disallow editing. But technological developments are increasing over time, the security of confidential messages in PDF format is perceived to be unsafe, as more and more applications can change or destroy messages in PDF format [1].

The MDTM Cipher algorithm stands for 'Monograph-Digraph-Transposition-Monograph' wherein the relationship between individual letters of plain text and ciphers is quite complex because each original letter is replaced by two letters each of which separately combines with the letters of the other pair to form Digraph, which is then re-converted into a monograph. This algorithm is part of a character-based classical algorithm that is very vulnerable to attack.

To provide dual security to the data specially in PDF file, this time the MDTM Cipher algorithm will be combined with the RC4A algorithm. The RC4A algorithm is a modified cryptographic byte-oriented stream cipher algorithm of RC4 ciphers whose security level is stronger than the usual RC4. The principal design principle of RC4A is to reduce the correlation between byte output and internal variables by making the byte output dependent on more random variables. This algorithm is part of a modern...
symmetric algorithm whereby modern algorithms are more known to be safe than classical algorithms. The technique of combining the two algorithms above is referred to as a super encryption method in which a classical symmetry algorithm combined with modern symmetric algorithms that the modern symmetric algorithm is expected to reinforce the previous algorithm.

2. Method
The RC4A algorithm is a symmetric key algorithm proposed by the FSE in 2004 by Paul S. and Bart Preneel which is a stream cipher modified by its predecessor RC4 algorithm [2]. The RC4A algorithm tries to improve RC4 security by using two arrays, there are $S_1$ and $S_2$ from 1 to 256 characters, two indexes $j_1$ and $j_2$, and two keys $k_1$ and $k_2$ corresponding to the Key Scheduling Algorithms (KSA) in Figure 1 and Pseudo Random Generation Algorithm (PRGA) process in Figure 2.

![Figure 1. KSA of RC4A](image-url)

![Figure 2. PRGA of RCA](image-url)
Whenever $i$ is added, two bytes are generated [3] with formula: $j_1 = S_1[j_1 + S[i] + key[ i \ mod \ key - length]] \ mod \ 256$ where the key is the input value of the user and the "key length" is the byte number of The key and range of the key length is from 1 to 256. For each iteration, values of $j_1$ and $j_2$ are calculated, $S_1[i]$ and $S_1[j_1]$ in swap (exchanged), and the sum of $S_1[i]$ and $S_1[j_1] \ mod \ 256$ After the result $S_1$ then done generate the first key and then for the calculation of $S_2$. These steps are repeated on $S_2$ and $j_2$, and $S[S_2[i] + S[j_2]]$ is the key stream output [4]. In the PRGA, the key generator (key stream) is XOR with plaintext to generate encryption text on the form based on the byte stream. The same operation applies to array $S_2$. Byte is returned, then XOR with one individual letter on plaintext to convert it to cipher text.

In MDTM Cipher algorithm formed a 16x16 square matrix. The sixteen lines are identified by the letters A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, and P. Then arrange the first plaintext into the box. Fill in the remaining boxes with unused characters from characters. If the keywords included in it are repeated, then they are ignored [5]. Each letter in the plaintext (called monograph) is now replaced with rows and column letters such that, for example, M becomes EM (called digraph). The resulting cipher text will be twice as long as the original. There are two types of keys used, the second key is a transposition key by inputting a number from 1 to 10 at random. The transposition keys are then used for the transposition process by arranging them into transposition tables and arranging them in order. The result is changed again to monograph in accordance with the first monograph table.

3. Results and Discussion

The results showed that the algorithm complexity obtained was quadratic with the result $\Theta (n^2)$. For the results of the processing time can be seen in Figure 3 and Figure 4 below.

![Figure 3. The Length of Plaintext Over Running Time Chart](image-url)
The picture above shows the plaintext and key plaintext graphs against the time of the encryption process, in which the length of the plaintext and the key is directly proportional to the processing time that produces a linear line. The linear line states that the longer the plaintext character and the key to be processed the longer the process time required.

4. Conclusion
As for the results of RC4A and MDTM Cipher algorithms implementation in the encryption of files on the Android platform, then the conclusion is:
1. The application designed in this research is capable of encryption and decryption using RC4A algorithm and MDTM Cipher shows that the final state array of the encryption and decryption process is the same.
2. The value of complexity of both algorithm used is RC4A and MDTM Cipher algorithm which consist of Key Scheduling Algorithm (KSA) and Pseudo-Random Generation Algorithm (PRGA) resulted $\Theta (n^2)$. This complexity is directly proportional to the user input so that the graph formed is a linear graph.
3. Test results at the time of the process obtained that the algorithm process time is directly proportional to the length of plaintext characters and keys. This means that the longer the plaintext used, the longer it will be The time required for encryption and decryption of plaintext and key.

References
[1] Dibiyo M S and Aisyatul K 2016 Teknik Penyembunyian Pesan PDF Terenkripsi Menggunakan Algoritma Kriptografi Vernam Cipher Dan Steganografi End Of File (Eof) Dalam Media Gambar (Semarang : Universitas Dian Nuswantoro).
[2] Paul S & Bart P 2004 A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher* Int. Workshop on Fast Software Encryption pp 245-259.
[3] Maximov A 2004 Two Linear Distinguishing Attacks on VMPC and RC4A and Weakness of RC4 Family of Stream Ciphers (Corrected) Sweden.
[4] Sarkar S 2010 Further Non-randomness in RC4, RC4A and VMPC (Chennai, India: Chennai Mathematical Institute).
[5] Churchhouse R F 2004 Codes and Ciphers Julius Caesar, the Enigma and the internet: Cambridge.