An Enhanced RC5 (ERC5) Algorithm Based on Simple Random Number Key Expansion Technique

Excel B. Villanueva, Ruji P. Medina
Technological Institute of the Philippines
Quezon City, Manila, Philippines
excel.villanueva.0816@gmail.com,
ruji_p_medina@yahoo.com

Bobby D. Gerardo
West Visayas State University,
Iloilo City, Philippines
bobby.gerardo@gmail.com

***Abstract*—RC5 algorithm is lighweight in nature because it has low memory and low power requirement which makes it suitable to be implemented in devices with limited power and memory supply. However, it suffers from slow encryption speed compared to other encryption algorithms. The main purpose of this paper is to enhance this algorithm to increase its encryption speed through a simple yet fast random number addition-then-append key expansion technique. The enhancement includes the generation of random number to be added to the generated key which will be repeated for two rounds and later to be appended, to produce a key material. Additional blocks and bitwise operations are also included in the enhancement. Results show that the enhanced RC5 (ERC5) algorithm positively outperforms the traditional RC5 algorithm and successfully increased its encryption speed.**

***Index Terms*— RC5 Algorithm, Random Number Addition-Then-Appen;, Key Expansion; Encryption Speed.**

I. INTRODUCTION

Over the past years, Twitter, Skype, LinkedIN, Drpbox, Facebook etc. suffered information compromise due to malicious attacks by so-called hackers [1]. These incidents/issues attracted attention from almost all organizations [2] claiming the importance of protecting user's information. User's information privacy, confidentiality and integrity are the main functions of an encryption algorithm [3]. These algroithms transform any sensitive input data from the user, say a password, pin code, biometrcs, into an unreadable text also known as cipher text.

However, because series or combinations of mathematical formula are embedded in these algorithm, most of the time, they are computationally expensive. They are computationally expensive in a sense that they demand both memory and power during the execution process. Given this problem, it is hard to implement encryption algorithm in devices with limited memory and power supply. In order to protect these devices, lightweight cryptography algorithms were fabricated or designed to perform security in such devices as good as the conventional algorithms [4]. Furthermore, studies show that in implementing lightweight algorithms, block ciphers perform better than others [5].

Among the block ciphers, RC5 algorithm is lighweight in nature because it has low memory and low power requirement which makes it suitable to be implemented in devices with limited power and memory supply [6] [7] [8]. This algorithm is authored by one of the developers of RSA Algorithm, Sir Rivest in the year 1994. Aside from its low memory and low power requirements, it also has other desirable properties such as adaptability and simplicity. It is adabtable because users can choose the level of security by adjusting its parameters such as *w*-word size, *r*-round number and *b*-variable length [6] [7] [9] [10].

However, like any other algorithms, RC5 algorithm also posesses weaknesses. These weaknesses include poor diffusion and confusion, slow encryption speed, and low security due to its weak keys [7] [8]. One of the limitations of RC5 algorithm is its low encryption speed compared to other encryption algorithms [11]. It is important to note that encryption speed should be kept in mind when fabricating or designing an algorithm because the encryption holds an important role in securing information hence should be fast enough to cope with real time operations [12]. The main purpose of this paper is to enhance the mentioned algorithm to increase its encryption speed through a unique and simple, yet fast random number addition-then-append key expansion technique.

This paper is organized as follows: Section 1 covers the introduction, Section 2 includes all literature that contributed in the formulation of this paper; Section 3 consists of all the aspects of the proposed algorithm; Section 4 and 5 state the methodology used and experimental results respectively; Section 6 and 7 covers the summary and conclusion, and future works of the authors respectively.

II. RELATED WORKS

Various researches have been conducted to solve the different weaknesses of the RC5 Algorithm. Different approaches have been suggested, from the most complex to the simplest. Here are the following researches which the authors reviewed to come up with their unique approach to solve the mentioned weakness of RC5, specifically its slow encryption speed.

Amin, et.al [7] and Faragallah [8] suggested the combination of RC5 algorithm and chaotic cryptosystem. They

used chaotic skew tent map to generate a round number instead of starting from a magic constant (which the traditional RC5 uses). Since they believe that combining chaotic cryptosystem can yield better pseudorandom property, it can increase the resistance of the RC5 algorithm to various attacks.

Like the previous researchers, Bajaj, et.al [13] also suggested a different approach to make RC5 algorithm more robust from attacks. This approach includes working on its number of rounds for encryption. It uses linear feedback shift register (LFSR) to calculate number of rounds, but take note that this number of rounds is only applicable to 1 block, meaning each block should have its own unique number of rounds. This approach was applied in an image encryption and proved its robustness.

Another simple approach was suggested by Gill, et.al [10] to enhance the performance of RC5 algorithm. This is very simple because it only uses a prime number as its round number. The authors believe that because prime numbers cannot be factored, it has been considered as building blocks in cryptography. The use of prime number in this approach is to increase its security. This approach was compared to a non-prime round number value and has proven its strength when it produced better result compared to the latter.

However, despite these researches, the authors feel that there is still a need to investigate further to solve the drawbacks/limitations of the classic RC5 algorithm. This led to the proposal of a simpler approach suitable to devices with limited supply, such as memory and power, hence this study. The next part of this paper will discuss the proposed algorithm.

III. Proposed Algorithm

The authors suggest an enhancement to solve one of the limitations of RC5 Algorithm which is its slow encryption speed. The enhancement includes the generation of random number to be added to the generated key which will be repeated for two rounds and later be appended to produce a key material. Additional blocks and bitwise operations were also included in the enhancement. Like the original RC5 algorithm, this enhancement also has 3 phases, 1) Key Expansion, 2) Encryption and 3) Decryption. This encryption can handle block sizes just like the classic RC5 such as 16, 32, 64 and 128-bit of plaintext and converts it into cipher text with the same size.

A. Enhanced Key Expansion Phase

For the key expansion phase, the authors made use of a uniquely fabricated key expansion technique, which they called random number addition-then-append key expansion technique. This enhancement omits the use of word size, array, and the use of magic constants that the algorithm originally have. This is a simple yet fast key expansion technique that involves three (3) simple steps which include, 1) Generation of random numbers, 2) Addition and 3) Append.

Figure 1 shows the diagram of this phase. As perceived in the diagram, the source key can be any random number that is based from the date and time. In order to expand this generated source key, the following steps will be executed: 1) hide the source key by simply generating random number to be added to the source key; 2) once generated, add that random number to each number in the source key, which is repeated twice (1st layer and second layer) and; 3) append the result of the two layers and at the end of the result, append the length of the original source key (1). To increase the confusion in this phase, it is important to note that the generated random number to be added in each layer is not equal or the same. The result of this phase is called key material, which will be used in the next phase which is the encryption phase.

$$KM = 1^{st}\ Layer \parallel 2^{nd}\ Layer \parallel L \quad (1)$$

where:
1st Layer = source key + random number
2nd Layer = 1st layer + new generated random number
L = Length of the source key

B. Encryption Phase

As a review, in the traditional RC5 Algorithm, only two blocks were included, and bitwise operations such as XOR, and left rotate [6]. Wherein this enhancement, with additional two blocks and additional bitwise operation, such as a combination of left and right rotate, 1's complement and XOR Swap were included. Figure 2 shows the encryption process which includes pre-processing of plaintext, division of plaintext, and block XOR swapping. To fully understand the different processes in the enhanced encryption phase, the authors discussed each process thoroughly below.

1. *Pre-processing of Plaintext*

Figure 3 shows the first process, the pre-process of plaintext. Any input from the user, be it a password or series of random numbers, will be converted into its hexadecimal value. At the same time, the key material, which is the result of the previous phase, the key expansion, will also be converted in its hexadecimal value. Both values will be XOR-ed to produce the value of the plaintext. So, the value of the plaintext is the XOR-ed values of the input from the user and the key material.

2. *Division of the Plaintext*

The plaintext is divided into two left division and right division. Each division has its own sets of bitwise operation to encrypt the data.

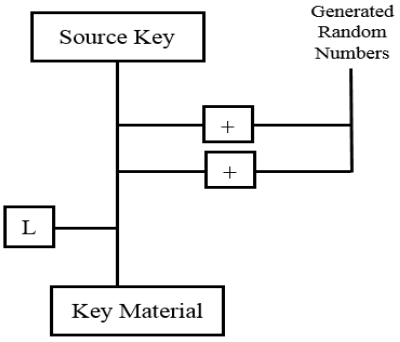

Fig. 1. Key Expansion Diagram