A Hybrid Approach for a Secured Information Security Using Modified Encryption Technique

Hanizan Shaker [1], Syazwani Yahya[2],
Munaliza Jaimun[3], Nik Zulkarnaen Khidzir[4]

[1] Senior Lecturer, Faculty of Computing and Engineering, Quest International University
[2] Lecturer, Faculty of Computing and Engineering, Quest International University
[3] Senior Lecturer, Faculty of Business and Management, Quest International University
[4] Senior Lecturer, Faculty of Creative Technology and Heritage, University Malaysia Kelantan

Abstract

This paper proposes a new approach called the RAES technique, which results from redesigning the current Rail Fence Cipher (RFC) using two basic phases, first using the Advanced Encryption Standard (AES) technique and then using the potential of the RFC technique to protect confidential messages for more secure information security. There are several conventional cryptographic methods, and because it is possible to crack cipher text, that is why it tries to suggest RAES techniques written in C++ programming to be more secure to protect information from cipher breaking. Mixing RFC ciphers with AES, it appears that the encryption and decryption of the modified RAES require the generation of the plaintext elements which are usually single letters written in a predetermined sequence into a matrix format which is basically a rectangle that has been decided by the transmitter and receiver in advance, and then it is read off according to another predetermined sequence across the matrix to get the cipher text. Through this RAES technique, not only the strength of the AES technique can be applied but also the RFC technique that uses keywords and salt can also be used making this mixed system perform ciphers that are difficult to break by attackers. Moreover, the strength of the RAES algorithm is in terms of faster and more secure execution times than existing substitution and transposition algorithms in addition to the improvement of confusion and diffusion characteristics. Meanwhile, the value of the avalanche effect for the RAES technique recorded also showed that it reached 60%.

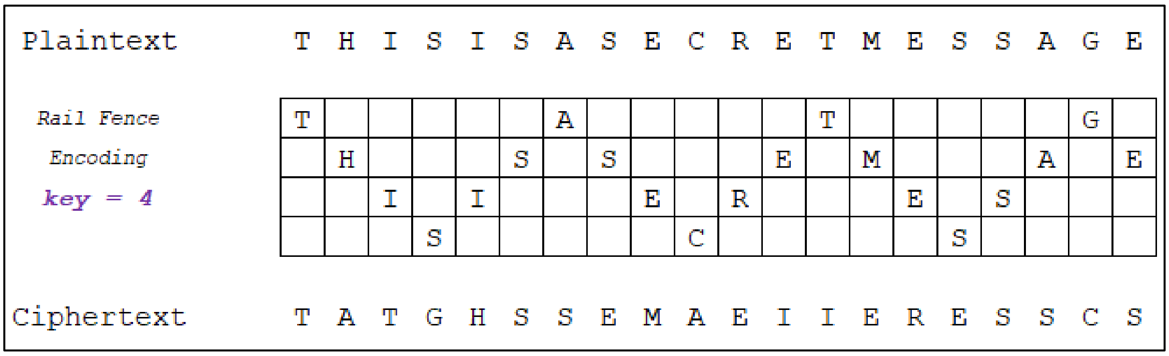
Keywords: cipher; encryption; plaintext; RFC; AES; RAES; modified encryption; information security

1. Introduction

The word ‘Cryptography’ according to Stallings (2014), is derived from a Greek word meaning secret writing. Cryptographic techniques are used to secure data. In cryptography plain text is transferred to cipher text using encryption techniques, this process is called encryption. And converting cipher text to plain text using decryption techniques, this process is called decryption. There are two main types of cryptographic techniques namely substitution and transposition techniques (Saini, 2015). In the substitution technique, plain text is replaced with numbers, symbols, and other characters. In contrast, transposition techniques involve rearranging letters in plaintext to encrypt a message, in which plaintext letters are replaced with letters from another alphabet or with different letters from the same alphabet (Aaref, 2017).

Many cipher techniques have been developed but the Rail Fence Cipher (RFC) is the simplest and most amusing cryptographic algorithm until now (Nahar & Chakraborty, 2020). RFC is a type of transposition technique. RFC is the enhanced version of Caesar Cipher. In the Caesar algorithm, the cipher text is derived by shifting each character in the plaintext by a certain value of key. Meanwhile, in RFC, the cipher text is obtained by writing plain text diagonally in a matrix. It was invented and used in the ancient times. The Greeks used it widely and also eventually made a scytale, which is a unique tool that made the encryption as well as decryption of secret message easier. All letters of the plaintext are organized in such a manner that resembles the shape of the top part of the rail fence with edges. It is also called a zigzag cipher because of the way it is encoded and moreover, it is an example of a transposition cipher. It falls under a class of transposition ciphers which is known as route ciphers which were famous in the early history of cryptology. In general, the plaintext elements which are usually single letters are written in a predetermined sequence into a format of matrix which is basically a rectangle that has been decided by the transmitter and receiver in advance, and then it is read off according to another predetermined sequence across the matrix to get the cipher (see Figure 1). The matrix, the starting point, and the sequences are all kept undisclosed in a route cipher (Akash et al., 2017).

Figure 1: Example of Rail Fence Cipher



Plaintext T H I S I S A S E C R E T M E S S A G E

Rail Fence
Encoding
key = 4

T						A						T						G	
	H				S		S				E		M				A		E
		I		I				E		R				E		S			
			S						C						S				

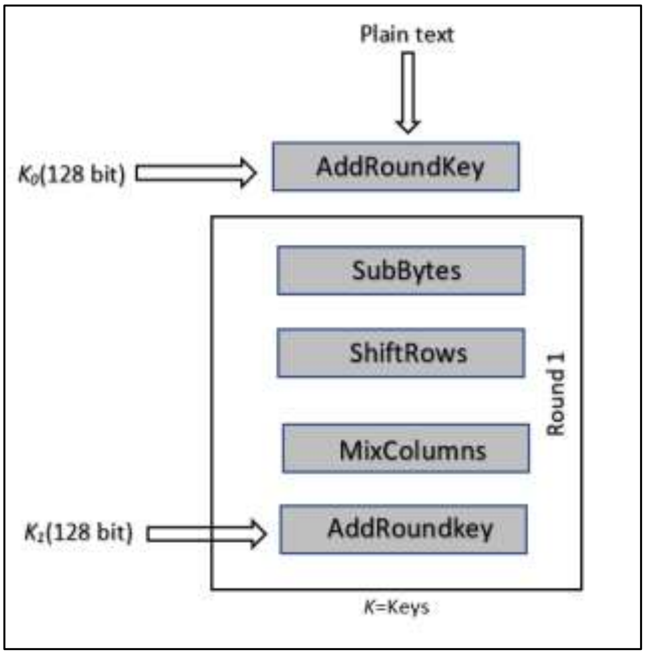
Ciphertext T A T G H S S E M A E I I E R E S S C S

But according to Godara et al., (2018) also stated that among the significant problems faced by the cipher text generated by the RFC algorithm tends to be easily decomposed using brute force attack, thorough search, search by frequency, and many other methods. This is because the secret information it encrypts is easy to reveal because the rate of diffusion and confusion is not strong and robust (Saini, 2015).

Meanwhile, there is another technique called Advanced Encryption Standard (AES). The AES algorithm is one of the symmetric key block digits with block sizes varying from 64 to 256 bits (Burr, 2003). AES has a 128-bit length that can encrypt and decrypt with three different key lengths as 128-bit, 192-bit and 256-bit known as AES128, AES192 and AES256 (Tillich et al., 2008). Rijndael is the name given to AES by its two creators, Belgian cryptographers Vincent Rijmen and Joan Daemen. The basic concept of the actual AES structures as described is shown in Figure 2.

AES is considered to be the strongest encryption standard because it cannot be broken by force even with current computing power. Even though AES can accept block sizes of up to 256 bits, its speed is still slow compared to flow-based ciphers at a time when all applications are looking for faster encryption processes such as web servers and Automatic Teller Machines (ATMs). On the other hand, some AES applications continue to struggle for low performance areas such as smart cards and cellular phone -related hardware. Therefore, encryption speed and execution times are two important factors for the real -time use of AES algorithms. The problem with the use of AES is the compromise between the speed of encryption and decryption and execution time where there is more confusion and diffusion. Furthermore, AES is said to use an algebraic structure which is too simple and each block is always encrypted in the same way that making it easy to break (Tillich et al., 2008; Stalling, 2017).

Figure 2: Basic Concepts of AES Structure



Therefore, this paper proposes a new approach called the RAES technique, which results from redesigning the current RFC using two basic phases, first using the AES technique and then using the potential of the RFC technique to protect confidential messages for more secure information security.

2. Related Studies

Umang Bhargaval et al., (2017) proposed a new algorithm that combines two techniques: Transposition and substitution. The proposed algorithm goes through three phases. The Substitution (Zero Multiplying) technique was used in the first phase. The output of the first phase is followed by a transposition technique (rail fence). The zero text (2nd phase output) is then replaced with the output producer symbol. While Akash et al., (2017), have introduced A.J.Cipher, using more than one encryption method. It uses a substitution cipher (Vigenere cipher) in the first stage. The first-level output along with the key is converted to ASCII-to-binary and executes the XOR operation. The output of this stage is converted back to binary-to-ASCII. ASCII equivalent characters are the final output.

Andysah Putera et al., (2016) redesigned the rail fence and suggested that the main letters of the message would be occupied in the bottom left -most corner of the matrix and then run from corner to corner upwards by placing the remaining letters. Then read the characters from the top row to the bottom. This generates an output message. Abitha et al., (2015) proposed a new cryptographic approach to maintaining privacy. They modified the ciphers: Rail fence and Vigenere cipher for data mining and used them to encrypt bank account numbers. Baljit Saini (2015) also presented an algorithm using cipher substitution (Modified Ceasar) and transposition cipher (Real Fence). It uses a modified Caesar technique in the first stage and the rail fence is the last stage.

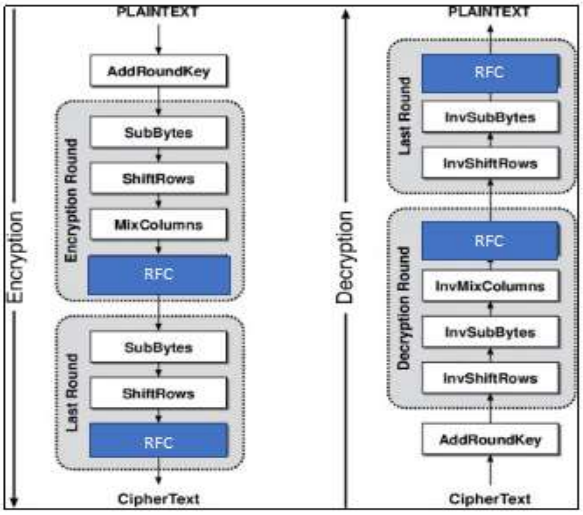
All techniques suggest two different levels of encryption where one of them uses the Real Fence technique. But no previous study has emphasized the confusion and diffusion to this Real Fence technique. Only recently, there have been few previous studies that emphasize on combining AES with other techniques that emphasize on increasing execution times. While Guru and Ambhaikar (2021) proposed a hybrid encryption algorithm mixing AES and RSA algorithms to overcome the file encryption performance and security problems. Meanwhile, Arman et al., (2020) introduced a new technique called AES-Symmetric Key Cryptosystem, with the aim of security and easy implementation. Lately, confusion and diffusion are something essential need to be achieved as mentioned in (Rosyidah et al., 2017).

3. Proposed Model

First of all, for RAES encryption, the plain text and initial key are added to the block using an XOR ("exclusive or") cipher, which is a built-in operation of the processor hardware. The plain text used is in the form of a text file where the characters used are between 10 to 50 characters as in (Aaref & Ablhd, 2017; Nahar & Chakraborty, 2020).

Then each byte of data is replaced with another, according to a predefined schedule. Next, the 4 × 4 array rows are moved: the bytes in the second row are moved one space to the left, the bytes in the third row are moved two spaces, and the bytes in the fourth row are moved three. The columns are then mixed by using the mathematical operation that combines four bytes in each column. Finally, the RFC algorithm is applied from the resulting blocks, and the process is repeated for each round. This results in a cipher text that is radically different from the regular text. For RAES decryption, the same process is performed in reverse (See Figure 3).

Figure 3: Main Construction of RAES Algorithm



The AddRoundkey step is implemented only once for the RAES technique to minimize the time taken for encryption and decryption operations. It is very important, short time while good data protection is a requirement for real time applications. Meanwhile, the steps such as SubBytes, ShiftRows and MixColumns are preserved to get more diffusion while the RFC process is to get more confusion.

The RFC algorithm process is further discussed in the next section and subsection.

4. Algorithm of RAES Techniques

4.1 RAES Encryption Algorithm

This algorithm is called the RAES encryption algorithm which takes place according to the following steps. It starts by reading the plain text and keywords for the AddRoundKey operation. Then it will go to the next step which is through the AES algorithm and to the last step which is the RFC algorithm before the cipher text is generated.

Steps	RAES Algorithm
i.	Read plain text.
ii.	Read the keyword.
iii.	Perform AddRoundKey using the XOR operation.
iv.	Perform SubBytes by replacing each byte of data with another according to a predefined schedule.
v.	Perform ShiftRows by moving the 4 × 4 array row, where the bytes in the second row are moved one space to the left, the bytes in the third row are moved two spaces, and the bytes in the fourth row are moved three.
vi.	Perform MixColumns by mixing columns using the mathematical operation of combining four bytes in each column.
vii.	Run the RFC algorithm from the blocks finally generating a text cipher for the following round.
viii.	IF Number of rounds ≠ Number of last rounds Go to Step iv to Step vii. ELSE Go to Step iv to Step v Then Go to Step vii
ix.	Save the resulting plain text as cipher text

4.2 RAES Decryption Algorithm

The REAS decryption algorithm takes place starting with reading the cipher text and keywords for the AddRoundKey operation. Then it will go to the next step which is to go through the AES algorithm and to the last step which is the RFC algorithm before the plain text is generated. The cipher text decryption process will take place where the steps as follows:

Steps	RAES Algorithm
i.	Read plain text.
ii.	Read the keyword.

iii.	Perform AddRoundKey using the XOR operation.
iv.	Perform SubBytes by replacing each byte of data with another according to a predefined schedule.
v.	Perform InvSubBytes by replacing each byte of data with another according to a predefined schedule.
vi.	Perform InvMixColumns by mixing columns using the mathematical operation of combining four bytes in each column.
vii.	Run the InvRFC algorithm from the blocks finally generating a text cipher for the following round.
viii.	IF Number of rounds ≠ Number of last rounds
ix.	Go to Step 4 to Step 7

5. Result

To get started, AES must first be executed with a plain text. For example, the plain text that used as input to AES:

AES plain text:	MEET ME AT QUEST INTERNATIONAL UNIVERSITY
Output cipher text:	U2FsdGVkX19eUOzxd4RrAxqv2moIDvCDgUSxTtjuVVtOx7sUW6H3JBXlHqMktyJYuwGeX5FaHJI3HkMMxuV8g

The result is strong and difficult to break and the same character is ciphered to different characters and all the plain text characters are quite different from the original text. The steps of AES that lead to the above results are as follows:

Steps: MEET ME AT QUEST INTERNATIONAL UNIVERSITY

Pass to AES algorithm, where used as follow:

Keyword = INTRODUCING YOU A NEW TECHNIQUE!

Salt = NEW CIPHER RAES TECHNIQUE

ADDROUNDKEY:

```
void addRoundKey(unsigned char* state, unsigned char* roundKey){
      for(int i=0; i<16; i++){
            state[i] ^=roundKey[i];
      }
}
```

SUBBYTES:

```
void subBytes(unsigned char* state){
      for(int i=0;i<16;i++){
            state[i]=s_box[state[i]];
```

```
        }
}
```

SHIFTROWS:

```
void shiftRows(unsigned char* state){
        unsigned char tmp[16];

        tmp[0]=state[0];
        tmp[1]=state[5];
        tmp[2]=state[10];
        tmp[3]=state[15];
.
.
.
.
tmp[15]=state[11];

        for(int i=0;i<16;i++){
                state[i]=tmp[i];
        }

}
```

MIXCOLUMNS:

```
void mixColumns(unsigned char* state){
        unsigned char tmp[16];
        // Column 1 entries
        tmp[0] = (unsigned char) (mul2[state[0]] ^ mul3[state[1]] ^ state[2] ^
state[3]);
        tmp[1] = (unsigned char) (state[0] ^ mul2[state[1]] ^ mul3[state[2]] ^
state[3]);
        tmp[2] = (unsigned char) (state[0] ^ state[1] ^ mul2[state[2]] ^
mul3[state[3]]);
        tmp[3] = (unsigned char) (mul3[state[0]] ^ state[1] ^ state[2] ^
mul2[state[3]]);
.
.
.
.

        // Column 4 entries
        tmp[12] = (unsigned char) (mul2[state[12]] ^ mul3[state[13]] ^
state[14] ^ state[15]);
        tmp[13] = (unsigned char) (state[12] ^ mul2[state[13]] ^
mul3[state[14]] ^ state[15]);
        tmp[14] = (unsigned char) (state[12] ^ state[13] ^ mul2[state[14]] ^
mul3[state[15]]);
```

```
        tmp[15] = (unsigned char) (mul3[state[12]] ^ state[13] ^ state[14] ^
mul2[state[15]]);

        for (int i = 0; i < 16; i++)
           state[i] = tmp[i];

}
```

Output (Cipher text):

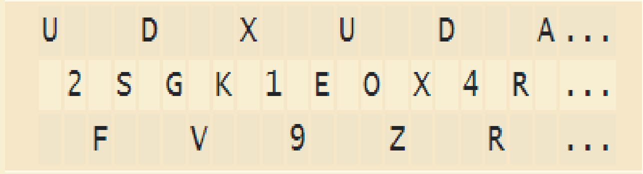
"U2FsdGVkX19eUOzxd4RrAxqv2moIDvCDgUSxTtjuVVtOx7sUW6H3JBXlHqMk-tyJYuwGeX5FaHJI3HkMMxuV8g"

Now, this cipher text is considered as plain text to the RFC algorithm by use the encryption key as follows:

- In the RFC algorithm, the order of the alphabet is re-arranged to obtain the cipher text.
- Where the plain text is written downwards and diagonally on successive rails of an imaginary fence.
- When the pointer reaches the bottom rail, it will traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus, the alphabet of the message is written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher text.

For example, if the message is:
“U2FsdGVkX19eUOzxd4RrAxqv2moDvCDgUSxTtjuVVtOx7sUW6H3JBXlHq
MtyJYuwGeX5FaHJI3HkMMxuV8g”
and key (number of rails) = 3 then cipher is prepared as in Figure 4.

Figure 4: Cipher text in RFC



```
U   D   X   U   D   A...
 2 S G K 1 E O X 4 R ...
  F   V   9   Z   R  ...
```

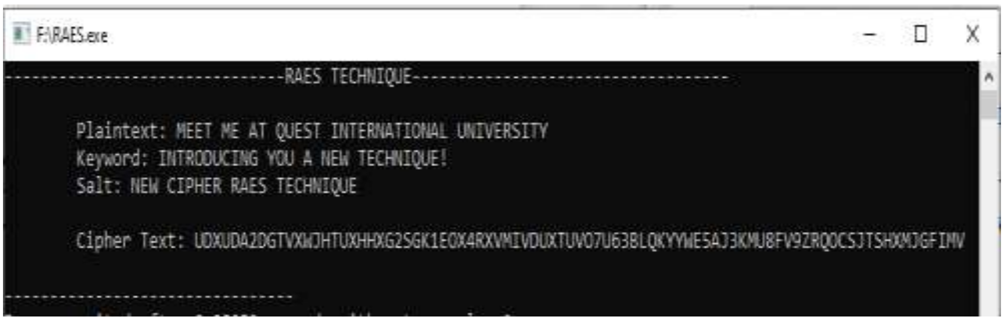
With the same way it gets the cipher text:
“UDXUDA2DGTVXWJHTUXHHXG2SGK1EOX4RXVMIVDUX-TUVO7U63BLQKYYWE5AJ3KMU8FV9ZRQOCSJTSHXMJGFIMV”

From the algorithm implemented above, can be summarized as follows:

Plaintext : MEET ME AT QUEST INTERNATIONAL UNIVERSITY
Ciphertext : UDXUDA2DGTVXWJHTUXHHXG2SGK1EOX
4RXVMIVDUXTUVO7U63BLQKYYWE5AJ3KMU8FV9ZRQOCSJTSHXMJGFIMV

Figure 5 shows the output from the implementation of the RAES Technique.

Figure 5: Output from the program using RAES technique



```
-----------------------------RAES TECHNIQUE----------------------------------

      Plaintext: MEET ME AT QUEST INTERNATIONAL UNIVERSITY
      Keyword: INTRODUCING YOU A NEW TECHNIQUE!
      Salt: NEW CIPHER RAES TECHNIQUE

      Cipher Text: UDXUDA2DGTVXWJHTUXHHXG2SGK1EOX4RXVMIVDUXTUVO7U63BLQKYYWE5AJ3KMU8FV9ZRQOCSJTSHXMJGFIMV


------------------------------
```

Meanwhile, the decryption function requires the use of the inverse of the whole process from beginning to end due to the symmetric technique. Table 1 shows the time taken with the use of the RAES Technique based on key types.

Table 1: Time taken based on key types

Keys	Time Taken
RAES 128	0.18
RAES 192	0.38
RAES 256	0.40

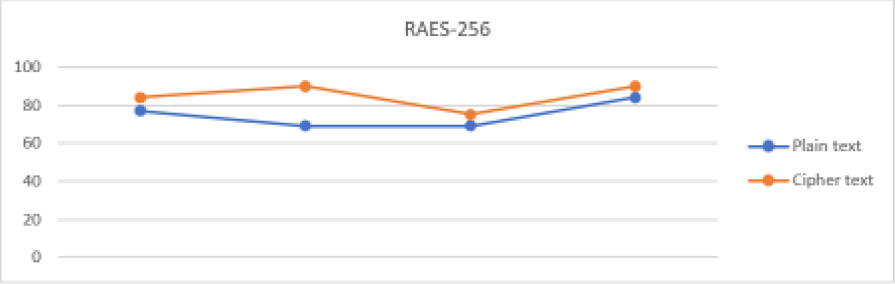
When using the RAES Technique to encrypt plain text similar to that used above, for key 128 (RAES 128) the time taken is 0.18 seconds. While for RAES 192 (key is equivalent to 192) recorded a time of 0.38 seconds and 0.40 seconds was recorded for RAES 256. It shows a longer time taken if encrypting the same plain text for RAES 256 compared to RAES 192 and RAES 128.

Figure 6 : Characters to ASCII numeric value converting table

M	E	E	T	…	U	N	I	V	E	R	S	I	T	Y
77	69	69	84	…	85	78	73	86	69	82	83	73	84	89

In the meantime, Figure 6, shows an example where each character for each text (i.e. for plain text and cipher text) is converted to an ASCII numeric value first as in (Rasyidah et al., 2017).

Figure 7 : Diffusion and confusion level from RAES-256 technique



Meanwhile, Figure 7 shows the observation for two values, namely plain text and cipher text after being converted to numeric values. Thus, the results of RAES-256 (for key 256) levels of diffusion and confusion proved to be higher when the graph above shows a significant difference between the lines representing plain text and cipher text. Here, an avalanche effect can be seen which is a method of measuring the effect on cipher text with respect to small changes made in plain or key text. A good encryption algorithm should always meet the following relationship i.e. the avalanche effect must pass more than half the changeover plain text (avalanche effect > 50%) (Shannon, 1949; Trappe & Washington, 2006). In order to determine the avalanche effect, the cipher text has to be converted to binary digits. Then the number of changed bit needs to be divided by the number of total bits in cipher text.

Avalanche Effect = (Number of Changed bit in ciphertext) / (Number of bits in ciphertext)

= 427 / 696

= 0.61 ≈ 61%

Table 2: Average changed in cipher text

	Avalanche Effect **(Average changed in cipher text)**
Rail Fence Only	32.5%
AES Only	51.4%
RAES	61.0%

Table 2 shows the avalanche effect values implemented using Rail Fence only, AES only and RAES. While the plain text and keyword used are the same. Rail Fences only showed an avalanche effect value of 32.5% while AES only recorded 51.4%. Meanwhile, the RAES technique recorded the highest avalanche effect value of 61.0%.

6. Discussion and Conclusion

There are several conventional cryptographic methods, and because it is possible to crack cipher text, that is why it tries to suggest RAES techniques written in C++ programming to be more secure to protect information from cipher breaking. Mixing RCF ciphers with AES, it appears that the encryption and decryption of the modified RAES require the generation of the plaintext elements which are usually single letters written in a predetermined sequence into a matrix format which is basically a rectangle that has been decided by the transmitter and receiver in advance, and then it is read off according to another pre-determined sequence across the matrix to get the cipher text. Through this RAES technique, not only the strength of the AES technique can be applied but also the RFC technique that uses keywords and salt can also be used making this mixed system perform ciphers that are difficult to break by attackers. Moreover, the strength of the RAES algorithm is in terms of faster execution times and more secure than existing substitution and transposition algorithms. It is faster because it uses Object Oriented programming such as C++ language.

In conclusion, it was found that originally, the cipher text generated by the RFC algorithm was prone to be easily decomposed using force, thorough search, search by frequency and many other methods as it had no diffusion and confusion in the generating algorithm. Similarly, to AES itself, it still has its short-comings such as speed is still slow especially AES256 but with a combination of RFC and modified AES known as RAES technique, adds a high percentage of confusion and diffusion in algorithms that generate a strong and difficult to crack ciphers.

References

1. Aaref, A. M. & Ablhd, A. Z. (2017). A New Cryptography Method Based on Hill and Rail Fence Algorithms. Diyala Journal of Engineering Sciences, 10(1), 39-47.
2. Arman, S., Rehnuma, T., & Rahman, M. (2020, December). Design and Implementation of a Modified AES Cryptography with Fast Key Generation Technique. In 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE) (pp. 191-195). IEEE.
3. Burr, W.E. (2003). Selecting the advanced encryption standard. IEEE Security and Privacy 1(2), 43–52.
4. Dar, J. A., Rafiq, M., & ul Rashid, F. (2015). Taxonomy of Changeover Ciphers Using Soft Computing Tools.
5. Godara, S., Kundu, S., & Kaler, R. (2018). An Improved Algorithmic Implementation of Rail Fence Cipher. International Journal of Future Generation Communication and Networking, 11(2), 23-31.
6. Guru, M. A., & Ambhaikar, A. (2021). AES and RSA-based Hybrid Algorithms for Message Encryption & Decryption. Information Technology in Industry, 9(1), 273-279.
7. Nahar, K., & Chakraborty, P. (2020). Improved Approach of Rail Fence for Enhancing Security.
8. Rosyidah, T., Imran, S., Lubis, Andi Marwan, E., Amir Mahmud H. & Harahap, M. (2017). A Simple Compression Scheme Based on ASCII Value Differencing. Journal of Physics: Conference Series, Volume 1007.
9. Saini, B. (2015). Modified Ceaser Cipher and Rail fence Technique to Enhance Security. International Journal of Trend in Research and Development, 2(5), 348-350.
10. Shannon, C. E. (1949). Communication Theory of Secrecy Systems. Bell System Technical Journal, vol. 28-4, pages 656–715, 1949.
11. Soni, J. K. & Soni, J. K. (2017). A.J.Cipher in 2nd International Conference on Telecommunication and Networks (TEL-NET 2017), Noida, India.

12. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice, Global Edition. Pearson. p. 177. ISBN 978-1292158587. ACM Woodstock conference
13. Stallings, W. (2014), Cryptography and Network Security Principles and Practices, Fourth Edition, Prentice Hall
14. Tillich, S., Feldhofer, M., Popp, T.& Großsch¨adl, J. (2008). Area, delay, and power characteristics of standard-cell implementations of the AES S-Box. Journal of Signal Processing Systems 50(2), 251–261.
15. Trappe, W. & Washington L. C. (2006). Introduction to Cryptography with Coding Theory. Second edition. Pearson Prentice Hall, 2006.