Analysis of AES-RC4 Encryption Based on Avalanche Effect

Reshma Chowdary Bobba
Department of Computer Engineering San Jose State University
reshma.chowdarybobba@sjsu.edu

Prashansa Evangeline Bonapalle
Department of Computer Engineering
San Jose State University
prashansaevangeline.bonapalle@sjsu.edu

Karthigai Amutha Ezhilarasu
Department of Computer Engineering
San Jose State University
karthigaiamutha.ezhilarasu@sjsu.edu

***Abstract*— With the exponential growth of data being transferred online, new approaches are urgently needed to protect sensitive user information. Securing sensitive data is an immense challenge as users increasingly communicate on public networks. Interception and theft of information transmitted online leads to compromised privacy and financial loss. Cryptography provides techniques to enhance data security, chiefly encryption to encode plaintext into ciphertext, and decryption to decode ciphertext back into plaintext. Algorithms grow vulnerable over time as computing power expands. This research develops a novel hybrid cryptography architecture to strengthen data protection. A combination of RC4 and the Advanced Encryption Standard (AES) algorithms [1], [2] is implemented for software encryption. The use of the avalanche effect tests the complexity of an algorithm and the strength of encryption is measured by flipping one key bit and comparing the original to the changed ciphertext. More difference proves better encryption. Our AES-RC4 hybrid [1], [2] flips 50.59% of bits when we tweak keys, because of the small file size but a higher difference can be achieved with large data files, which shows that AES-RC4 [1], [2] avalanche works better than AES or RC4 alone.**

Keywords*— *Encryption, Decryption, AES, RC4, Security, Hybrid Algorithm, Avalanche Effect.

I. Introduction

In today's digital world, safely transferring and storing data is a challenging task. The importance of data encryption cannot be overstated. Encryption protects sensitive information from unauthorized users. As our lives depend more and more on interconnected devices, for all our day-to-day activities including financial transactions, home security, healthcare, etc the need to protect digital assets has become paramount.

Encryption is a technique that transforms plain text into an unreadable form of information which is called ciphertext. The transmitter encrypts the message and transfers that piece of information through the non-secure channel. Even if a malicious person intercepts this ciphertext, they are not able to retrieve the original message without the knowledge of the encryption technique and the key used for the encryption. At the receiver end the message will be decrypted by the intended recipient using decryption technique and the appropriate key. This process makes sure that the confidentiality of the data is preserved.

The strength of the encryption algorithm depends on various factors including the key size, the complexity of the algorithm, the computational power to crack the encryption, etc. Many encryption algorithms, once strong enough, now have many vulnerabilities due to drastic improvements in the processing power of modern systems. This is the main reason for the rapid and ever-changing evolution of cryptography.

This paper aims to evaluate a hybrid encryption technique which is a combination of AES and RC4 encryption algorithms [1], [2]. Both AES and RC4 are symmetric Algorithms. The main difference between them is that AES is a block cipher while RC4 is a stream cipher. Both have their advantages and disadvantages. In this paper, we are going to analyze how combining these two algorithms will improve the Avalanche Effect(AE) of the cipher text. AE is a preferable characteristic of cipher text which defines how a small change in input text or key will affect the cipher text.

The rest of this paper is organized as follows: Chapter 2 describes the AES and RC4 algorithm and its steps, Chapter 3 talks about the proposed methodology, and Chapter 4 evaluates the experimental results.

II. Background Knowledge

The two algorithms we used in this experiment are AES and RC4. This section describes the two algorithms in detail.

A. AES Algorithm

The Advanced Encryption Standard (AES) [3] is a symmetric key cipher that encrypts 128-bit blocks with 128, 192, or 256-bit keys.

The plaintext input is copied into a 4x4 byte state array. It can be observed from Fig. 1 that each encryption round involves substituting bytes using a substitution table, shifting rows of the state array, and mixing the data in columns using a linear transformation.

1. SubBytes - A non-linear substitution cipher using a

16x16 lookup table called the S-box. Each byte of the block is substituted using an S-box [3].

2. ShiftRows - A simple transposition cipher that does a circular left shift of the last 3 rows of the state matrix by offsets of 1, 2, and 3 bytes respectively. Provides diffusion across rows [3].
3. MixColumns - A linear mixing operation that combines bytes of each column using matrix multiplication. Gives more diffusion [3], [4].
4. AddRoundKey - A simple XOR operation between the current state and a round key derived from the original encryption key using key expansion [3], [5].

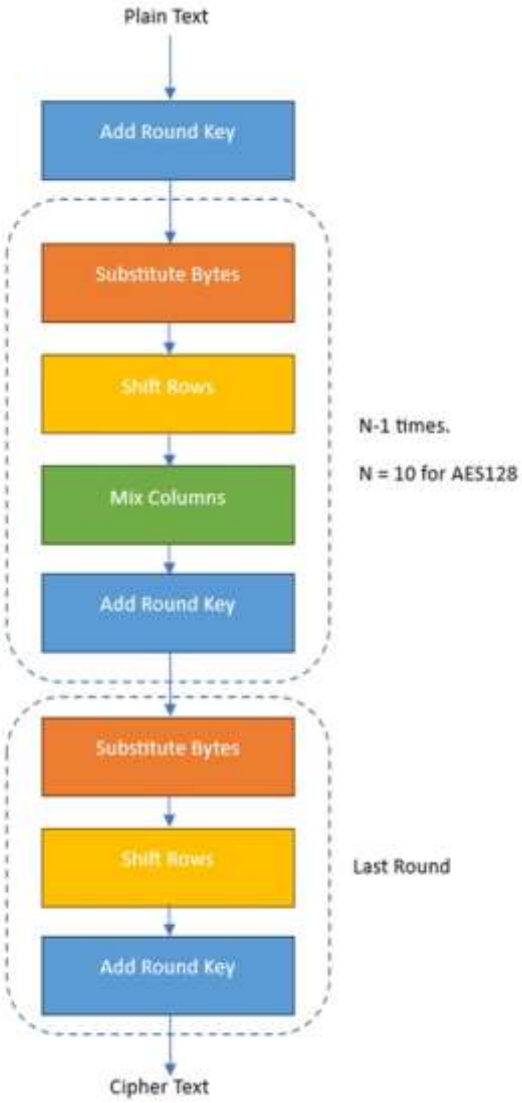

Fig. 1. AES Encryption Flow

So here, each AES encryption round applies four transformations using the secret key to thoroughly scramble the plaintext, and the inverse functions are used in reverse order for decryption.

B. RC4 Algorithm

RC4 is a symmetric stream cipher that uses a variable length key from 1 to 256 bytes [6]. As shown in Fig. 2, the algorithm has two phases:

1. Key Scheduling Algorithm (KSA): Uses a secret key to perform a permutation of a 256-byte state table S. Bytes of S are swapped iteratively based on the key [6].
2. Pseudo-Random Generation Algorithm (PRGA): Modifies state table S byte-by-byte to generate a keystream, completely independent of the plaintext, which is XORed with the plaintext to produce ciphertext [6].

Decryption generates the same keystream using the same key and XORing it with the ciphertext to recover the plaintext.

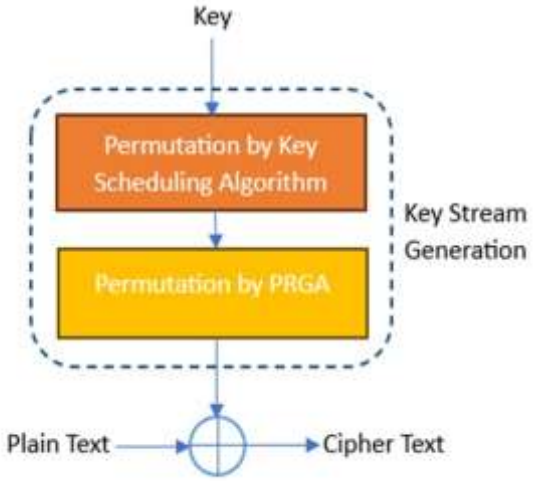

Fig. 2. RC4 Encryption

Here's a comparison of AES and RC4 algorithms [7] in Table 1.

TABLE I

COMPARISON OF AES AND RC4

AES	**RC4**
Block Cipher	Stream Cipher
Fixed Length (128, 192 or 256)	Variable Length (1 – 256 bytes)
Slow	**Fast**
Symmetric	Symmetric

Resource Intensive	**Light Weight**
Highly Secure	Vulnerable to key-related attacks

AES provides robust security by using uniform building blocks of fixed sizes to encrypt large files. This creates detectable patterns that may expose secrets making the block style a drawback even if the encryption is strong.

Whereas RC4 uses a fast pseudorandom number generator to encrypt data making it more vulnerable to key-related attacks [7]. By combining these two algorithms, a hybrid AES-RC4 [1], [2] approach is proposed that uses the security strengths of AES and the versatility and speed of RC4. Together, they result in a flexible and high-performance encryption solution.

III. Proposed Methodology *A.*

Encryption Scheme:

First, an AES ciphertext is created by encrypting the plaintext with an AES-128 key (128 bits). Subsequently, the AES ciphertext is encrypted with RC4 using the same key (128 bits/16 bytes). The final ciphertext, which has undergone this combined encryption procedure, is tested for the avalanche effect to determine the strength and quality of the encryption. The goal of this technique is to strengthen data safety by utilizing the complementary qualities of both the RC4 and AES algorithms [1]. *B. Decryption Scheme:*

Both RC4 and AES are used in the decryption technique to carry out the encryption procedure in reverse. The decryption procedure proceeds in reverse order if AES encryption and RC4 encryption were used in the original encryption process. In this instance, the ciphertext is first decrypted using the RC4 key, resulting in a plaintext that is still encrypted using AES. After that, AES decryption is performed using the AES key, which finally yields the original plaintext. In Fig. 2, the decryption procedures are shown. A bit error ratio (BER) test is performed to verify the integrity of the decrypted plaintext and guarantee its equivalency to the original. A complete evaluation of the encryption quality is made possible by this extensive decoding procedure [1].

$$BER = \frac{different\ bits\ of\ original\ and\ decrypted}{total\ bits\ of\ original\ file}$$

The Algorithms are implemented in C++. The input is a .txt file of size 1968 bits. As described in the methodology we are using the same key for both AES and RC4 Encryption. The AES version we implemented is AES128 which uses a 128-bit key. RC4 accepts various key lengths, in our case we used the 128-bit key. The two keys we used for the comparison are shown in Table 1. The keys are stored as arrays of uint8_t of length 16 (16 * 8 = 128 bits).

TABLE Ⅱ

KEYS USED

Key	**Key Value**	**Ascii Value of Changed Char**
Key 1	privatekey**1**23456	'1' - 0011 000**1**

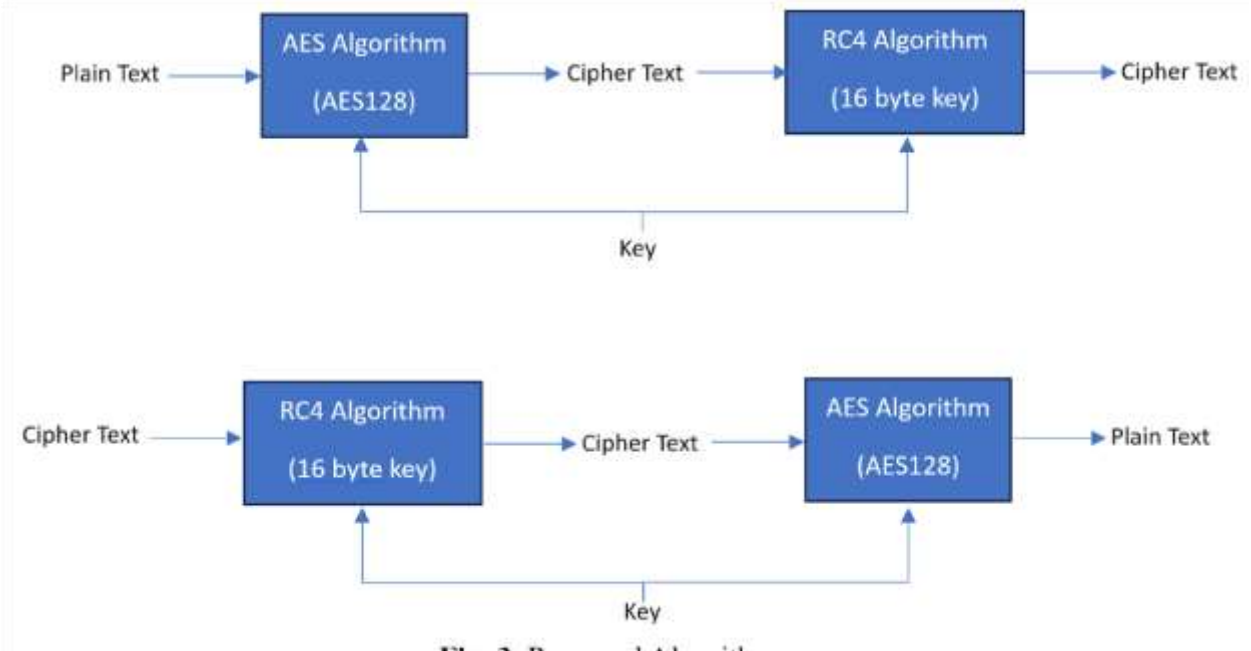

Fig. 3. Proposed Algorithm

Key 2	privatekey**0**23456	'0' - 0011 000**0**

To evaluate the proposed system the first metric we used is the Avalanche Effect. Avalanche effect (AE) is the measure of how the small change in the plain text or the key affects the cipher text. The equation for the avalanche effect is as follows

$$AE(cipher1, cipher2) = \frac{No\ of\ diff\ bits\ in\ new\ cipher\ text}{Total\ no\ of\ bits\ in\ cipher\ text}\ x\ 100\%$$

Cipher1 and cipher2 are the encrypted versions of the input text file with key1 and key2 respectively. The AE is the ratio of the number of different bits comparing the cipher1 and cipher2 to the total number of bits in the ciphertext. The Avalanche effect of AES, RC4, hybrid AES-RC4 [1], and hybrid AES-RC4 with reduced AES rounds [2] are shown in Table 2. The cipher text of AES has some extra bits in addition to 1968 input bits, this is because AES works on the data in terms of blocks of 16 bytes, and since the input is not a multiple of 16 bytes it is appended with some dummy bytes at the end. For the AE computation, we neglected those appended bits and considered only the first 1986 bits.

TABLE Ⅲ

AVALANCHE EFFECT COMPARISON

Algorithm	**Avalanche Effect**
AES	49.22%
RC4	47.7%
AES-RC4	**50.59%**
AES– RC4 (reduced Rounds) - 6 rounds	49.90%

As you can see the AES-RC4 hybrid algorithm [1], [2] has a better AE compared to encrypted using just AES and RC4 alone. As shown in Table Ⅲ, the hybrid method increases the encryption time. To overcome this effect we experimented with reducing the number of rounds in AES. This reduces the average time for encryption to 1881µs which is even less than the time taken for AES alone while maintaining the AE close to 50%.

TABLE Ⅳ

ENCRYPTION TIME COMPARISON

Algorithm	**Average Time for Encryption**
AES	2303µs
RC4	186µs
AES-RC4	2452µs
AES– RC4 (reduced Rounds) - 6 rounds	**1881µs**

Thus, a better trade between the AE and the encryption time is obtained by reducing the number of rounds in AES in the hybrid approach. We were able to obtain close to 50% AE with an 18% reduction in the time of encryption compared to AES and a 23% reduction in the time of encryption compared to the AES-RC4 hybrid method [1], [2].

V. Conclusion

In order to achieve enhanced security, this study combines the cryptographic techniques of RC4 and AES. The combined strategy is effective as demonstrated by empirical testing, which shows good performance. The combination of AES and RC4 achieves an impressive 50.59% in the avalanche test results, outperforming other methods. This elevated result suggests that the modified key's bit values have been changed effectively. As a result, the combination of the AES and RC4 algorithms shows that it can improve file encryption security.

VI. Acknowledgment

We would like to express our sincere gratitude to Professor Ammar Rayes who gave us immense insights about modern security techniques and encouraged us to do this project. Through this project, we gained valuable knowledge about various cryptographic techniques and algorithms.

VII. References

[1] N. Atikah, M. R. Ashila, D. R. Ignatius Moses Setiadi, E. H. Rachmawanto and C. A. Sari, "AES-RC4 Encryption Technique to Improve File Security," 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICIC47613.2019.8985825.

[2] P. K. Ketan and V. Vijayarajan, "An Amalgam Approach using AES and RC4 Algorithms for Encryption and Decryption," Int. J. Comput. Appl., vol. 54, no. 12, pp. 29–36, Sep. 2012.

[3] zeroFruit, "What is AES? — Step by Step - zeroFruit," Medium, Feb.13, 2019. https://zerofruit-

web3.medium.com/what-is-aes-step-by-step
fcb2ba41bb20 (accessed Nov. 15, 2023).

[4] Wikipedia contributors, "Rijndael MixColumns," Wikipedia, The Free Encyclopedia, Oct. 10, 2023. https://en.wikipedia.org/w/index.php?title=Rijndael_MixColumns&oldid=1179539921

[5] Neso Academy, "AES Key Expansion," Aug. 28, 2023. https://www.youtube.com/watch?v=0RxLUf4fxs8 (accessed Nov. 14, 2023).

[6] Wikipedia contributors, "RC4," Wikipedia, The Free Encyclopedia, Oct.29,2023. https://en.wikipedia.org/w/index.php?title=RC4&oldid=1182489702

[7] U. U. Follow, "Difference between RC4 and AES," GeeksforGeeks, Jul.21, 2023. https://www.geeksforgeeks.org/difference-between-rc4and-aes/ (accessed Nov. 14, 2023).