Journal of Physics: Conference Series

RBJ25 cryptography algorithm for securing big data

To cite this article: S Rajaprakash *et al* 2020 *J. Phys.: Conf. Ser.* **1706** 012146

RBJ25 cryptography algorithm for securing big data

S Rajaprakash[1], C Bagath Basha[2], S Muthuselvan[3], N Jaisankar[4] and Ravi Pratap Singh[5]

[1,2,3,5] Department of Computer Science and Engineering, Aarupadai Veedu Institute of Technology, Vinayaka Mission's Research Foundation, Chennai, Tamil Nadu, India.
[4] Department of Electronics Communication and Engineering, Misrimal Navajee Munoth Jain Engineering College, Chennai, Tamil Nadu, India.

Email: sravipratap56@gmail.com

Abstract. Cryptography is essential tool to protect data, information in computing system. In worldwide, billions of people using this cryptography methods in daily basis because It used the protect data. By and large the size of information imparted in web applications is developing and is in Peta-bytes and Exa-bytes. The customary security frameworks can't give successful insurance and confirmation to these sorts of informational collections. To over this issue in this work we have proposed a new cryptography algorithm RBJ25 with the matrix order N. This proposed algorithm contains three parts. In the first part based on the column operation of the matrix. In the second part secrete key generation and in the last part has five sub part to protect the information which will be discuss in detail in implementation part. Similarly in the decryption has three reverse parts. Finally, this algorithm is implemented and compared with traditional algorithms like AES and ChaCha20.

Keywords: ChaCha, Decryption, Encryption, Prime, RBJ25

1. Introduction

Today's data need more security is very important to all the areas in the world. For example, bank analysed data, social media analysed data, personal storage data, credit and debit card analysed data, and machine learning algorithm prediction data [13]. Now we discuss and find the various attacks and how this attacks are used. The first method discuss about fault attack of ChaCha, and it is used to rotate the XOR. The second method discuss about freestyle method, and it is used to different texts are key, nonce, and messages. The third method analysed the bricklayer attack. The fourth method attack is fault injection attack. This attack used to counting the block and added the matrix. The fifth method attack is hash function Double A. This function has two processes such as column process and row process. The sixth method attack is ann addition rotation XOR provide the high security. These methods are existing attacks of this paper. Finally, the final method is ChaCha20/4 process, and it is quarter round process and each process is southeast diagonal process, and provide tiny bit data security. To overcome this drawback, to proposed new algorithm Rajaprakash Bagathbasha Jaishankar25 (RBJ25) in this current work.

2. Related work

This part should contain sufficient detail so that all This author talks about fault attack of the additional rotations XOR for ChaCha family. This family analysis the variance fault attacks on ChaCha without resorting of the data [1]. Author introduced the method of Freestyle. This method has used different cipher texts are key, nonce, and message, and also introduced the new concept is hash based halting conditions and key guessing [2]. They analysis the side channel analysis for ChaCha. This analysis used to leakages related to the accesses of memory, and they are introduced the bricklayer attack [3]. They proposed fault injection attack on ChaCha and Salsa20 ciphers. This ciphers used to initialize the matrix, make a key, counting the block, nonce, and added the matrix [4]. This author discussed the hash function Double A. This function has two rounds. The first round is column round and it has three process. The first process is addition, second process is rotation of constant, and rotated XOR. The second round is row round and it has three process. The first process is addition, second process is rotation of constant, and rotated XOR [5]. They discussed about Salsa20 stream ciphers is ann addition rotation XOR. This cipher used for high security [6]. This author analysis the “Double A hash function” of the security for Salsa20 [7]. They analysis the power analysis attack and correlations power analysis for the vulnerability of Salsa20. The best attack is power analysis attack [8]. They are mainly studied the design and implementation of constant time web assembly. This design is fast and flexible to implement the secure algorithms [9]. They generalize the notion of probabilistic neutral bits to probabilistic neutral vectors, and the set of probabilistic neutral vectors is no smaller than that of probabilistic neutral bits. It is used to find and improved the key recovery attacks on reduced the round of Salsa20 and ChaCha [10]. SRB21 methodology are proposed by Somasundaram Rajaprakash Bagahbasha21 and they mainly discuss the prime numbers of the secret key [11]. SRB18 is mainly discuss about the protection of the twitter analysed data [12].

3. Methodology

The proposed methodology RBJ25 with the matrix of order N by N. The proposed algorithm has three processes. The first process is column operations in the matrix. The second process is secret key multiplication in the matrix. The third process has five processes. The first process is analyze the possible prime numbers in the given matrix. The second process is apply the possible prime number in quadratic equations in the given matrix. The third process is to merge all numbers into a single row. The fourth process is to form a pair from left to right side from third process. The fifth process is to swap the cell values with the help of pair from the given matrix.
RBJ25 algorithm

3.1. RBJ25 algorithm

3.1.1. RBJ25 encryption algorithm

RBJ25 encryption algorithm consists of eight steps.

Step 1:

$$CA = C_i \leftrightarrow \left(C_{i+(n-m)}\right)$$

$$\textit{Where } CA \textit{ is column encrypted matrix,}$$

$$C \textit{ is columns}, i, n \,\&\, m \textit{ is column numbers} \tag{1}$$

Step 2: To multiply the secret key in the matrix **A.**

$$\mathbf{A} = \mathrm{ek}.\mathbf{A} \tag{2}$$

where A is matrix, ek is encryption key.

Step 3: To analyse the possible prime number in the given matrix.

Step 4:

$$EM = \langle -b \pm \sqrt{b^2 - 4ac}\,\rangle / 2a$$

$$\textit{Where } EM \textit{ is encrypted matrix,}$$

$$a, b, \textit{and } c \textit{ are possible prime numbers} \tag{3}$$

Step 5: To merge all the numbers into a single row.

Step 6: To form a pair of numbers from left to right from Step 5.

Step 7: Each and every pair should swapped cell values from the given matrix.

3.1.2. RBJ25 decryption algorithm

RBJ25 decryption algorithm consists of seven steps.

Step 1: To analyse the possible prime number in the given matrix.

Step 2:

$$DM = \langle -b \pm \sqrt{b^2 - 4ac} \rangle / 2a$$
$$Where\ DM\ is\ decrypted\ matrix,$$
$$a, b, and\ c\ are\ possible\ prime\ numbers \qquad (4)$$

Step 3: To merge all the numbers into a single row.

Step 4: To form a pair of numbers from right to left from Step 3.

Step 5: Each and every pair should swapped cell values from given matrix.

Step 6: To divide the secret key in the matrix **A.**

A = A/dk (5)

where A is matrix, dk is decryption key.

Step 7:

$$CA = C_i \leftrightarrow \left(C_{i+(n-m)}\right)$$
$$Where\ CA\ is\ column\ encrypted\ matrix,$$
$$C\ is\ columns, i, n\ \&\ m\ is\ column\ numbers \qquad (6)$$

3.2. Working of RBJ25 encryption algorithm

- The propose RBJ25 algorithm developing from RB20 method.

$$\text{A=}\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

where A is analysed data matrix.

- By applying equations “(1)”

$$\text{A=}\begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$$

- By applying equations “(2)” and ek=5

$$\text{A=}\begin{bmatrix} 15 & 10 & 5 \\ 30 & 25 & 20 \\ 45 & 40 & 35 \end{bmatrix}$$

- By applying equations “(3)”

Step 1: a=2, b=3, c=7

Step 2: $EM = (-3 \pm \sqrt{3^2 - 4*2*7})/2*2$

$$EM = (-3 \pm \sqrt{9-56})/4$$
$$EM = (-3 \pm \sqrt{47})/4$$
$$EM = (-3 \pm 6.85565)/4$$

Step 3: Finally, EM = 36855654

Step 4: Pair of numbers (3, 6), (8, 5), (5, 6) and (5, 4).

Step 5: The 1st pair number (3, 6) should be swapped in the given matrix and this matrix represented start from 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9th cell number is 9-1.

$$\text{FPN=}\begin{bmatrix} 15 & 10 & 5 \\ 45 & 25 & 20 \\ 30 & 40 & 35 \end{bmatrix}$$

where FPN is first pair number

- The 2nd pair number (8, 5) should be swapped from FPN matrix.

$$\text{SPN=}\begin{bmatrix} 15 & 10 & 5 \\ 45 & 25 & 35 \\ 30 & 40 & 20 \end{bmatrix}$$

where SPN is second pair number

- The 3rd pair number (5, 6) should be swapped from SPN matrix.

$$\text{TPN=}\begin{bmatrix} 15 & 10 & 5 \\ 45 & 25 & 30 \\ 35 & 40 & 20 \end{bmatrix}$$

where TPN is third pair number

- The 4th pair number (5, 4) should be swapped from TPN matrix.

$$\text{FoPN=}\begin{bmatrix} 15 & 10 & 5 \\ 45 & 30 & 25 \\ 35 & 40 & 20 \end{bmatrix}$$

where FoPN is fourth pair number

- Finally, the original matrix could be encrypted successfully.

3.3. Working of RBJ25 decryption algorithm

$$\textbf{A=}\begin{bmatrix} 15 & 10 & 5 \\ 45 & 30 & 25 \\ 35 & 40 & 20 \end{bmatrix}$$

where A is encrypted data matrix.

- By applying equations "(4)"

Step 1: a=2, b=3, c=7

Step 2: $EM = (-3 \pm \sqrt{3^2 - 4 * 2 * 7})/2*2$

$EM = (-3 \pm \sqrt{9-56})/4$

$EM = (-3 \pm \sqrt{47})/4$

$EM = (-3 \pm 6.85565)/4$

Step 3: Finally, EM = 36855654.

Step 4: Pair of numbers (4, 5), (6, 5), (5, 8), and (6, 3).

Step 5: The 1st pair number (4, 5) should be swapped in the given matrix and this matrix represented start from 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9th cell number is 9-1.

$$\text{FPN=}\begin{bmatrix} 15 & 10 & 5 \\ 45 & 25 & 30 \\ 35 & 40 & 20 \end{bmatrix}$$

where FPN is first pair number

- The 2nd pair number (6, 5) should be swapped from FPN matrix.

$$\text{SPN}=\begin{bmatrix} 15 & 10 & 5 \\ 45 & 25 & 35 \\ 30 & 40 & 20 \end{bmatrix}$$

where SPN is second pair number

- The 3rd pair number (5, 8) should be swapped from SPN matrix.

$$\text{TPN}=\begin{bmatrix} 15 & 10 & 5 \\ 45 & 25 & 20 \\ 30 & 40 & 35 \end{bmatrix}$$

where TPN is third pair number

- The 4th pair number (6, 3) should be swapped from TPN matrix.

$$\text{FoPN}=\begin{bmatrix} 15 & 10 & 5 \\ 30 & 25 & 20 \\ 45 & 40 & 35 \end{bmatrix}$$

where FoPN is fourth pair number

- By applying equations “(5)” and dk=5

$$\text{FiPN}=\begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$$

where FiPN is fifth pair number

- By applying equations “(6)”

$$\text{FiPN}=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$\text{A}=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Finally, the encrypted matrix could be decrypted successfully.

4. Performance analysis

The performance analysis of the proposed RBJ25 cryptographic encryption algorithm compared with Traditional AES algorithm and existing ChaCha algorithm as shown in the Table 1.

Table 1. RBJ25 Encryption Performance

File size in bytes	ChaCha	AES	RBJ25
24	1.69	1.225	2.2
76	1.29	1.852	2.6
111	1.09	2.858	3.4
312	2.73	1.764	4.5
822	2.64	3.179	5.3
1531	3.4	2.039	5.5
6580	2.27	3.221	6.8

Table 1, the performance of the RBJ25 encryption algorithm compared with salsa and AES. The figure 1 shows the performance analysis of the RB25 encryption algorithm presented when compared with ChaCha and Traditional AES algorithms.

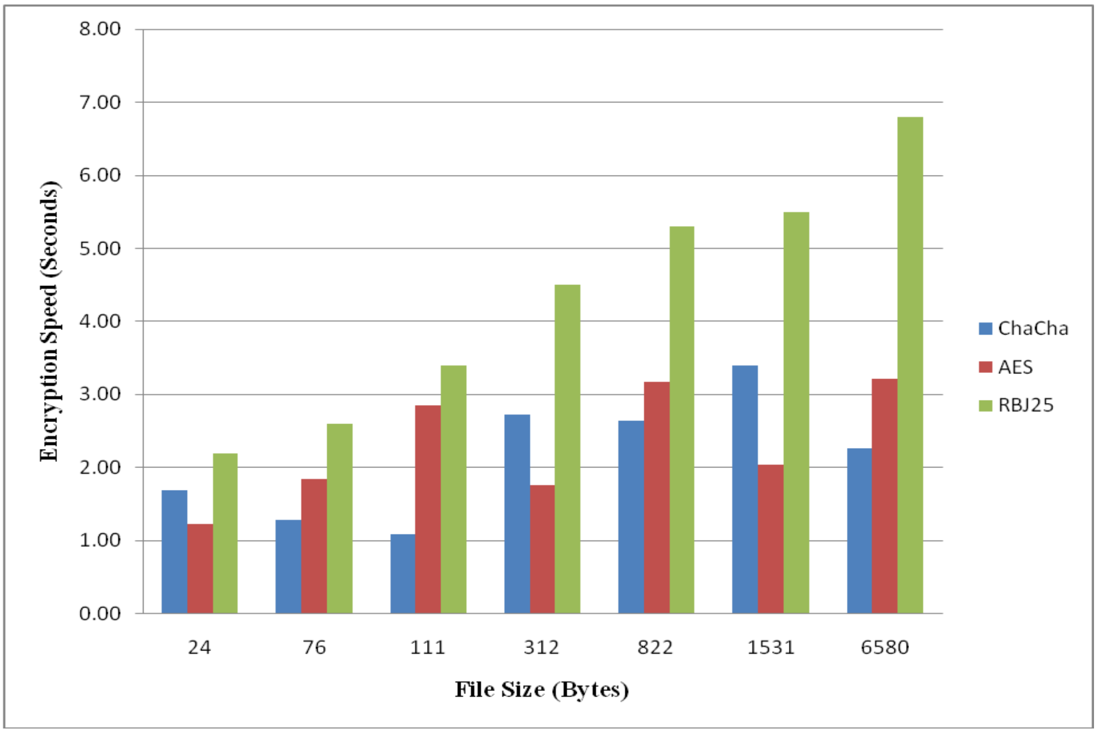

Figure 1. Encryption speed of RBJ25 algorithm compared with ChaCha and AES

- The performance analysis of the proposed RBJ25 cryptographic decryption algorithm compared with Traditional AES algorithm and existing ChaCha algorithm has been presented in the Table 2.
- Table 2, the performance of the RB25 encryption algorithm compared with ChaCha and AES.

Table 2. RBJ25 Decryption Performance

File size in bytes	ChaCha	AES	RBJ25
24	1.66	1.21	2.12
76	1.25	2.12	2.47
111	1.06	2.84	3.29
312	2.64	1.72	4.44
822	2.59	3.17	5.18
1531	3.34	2.29	5.28
6580	2.21	3.20	6.68

- The figure 2 shows the performance analysis of the RB25 decryption algorithm presented when compared with ChaCha and Traditional AES algorithms.

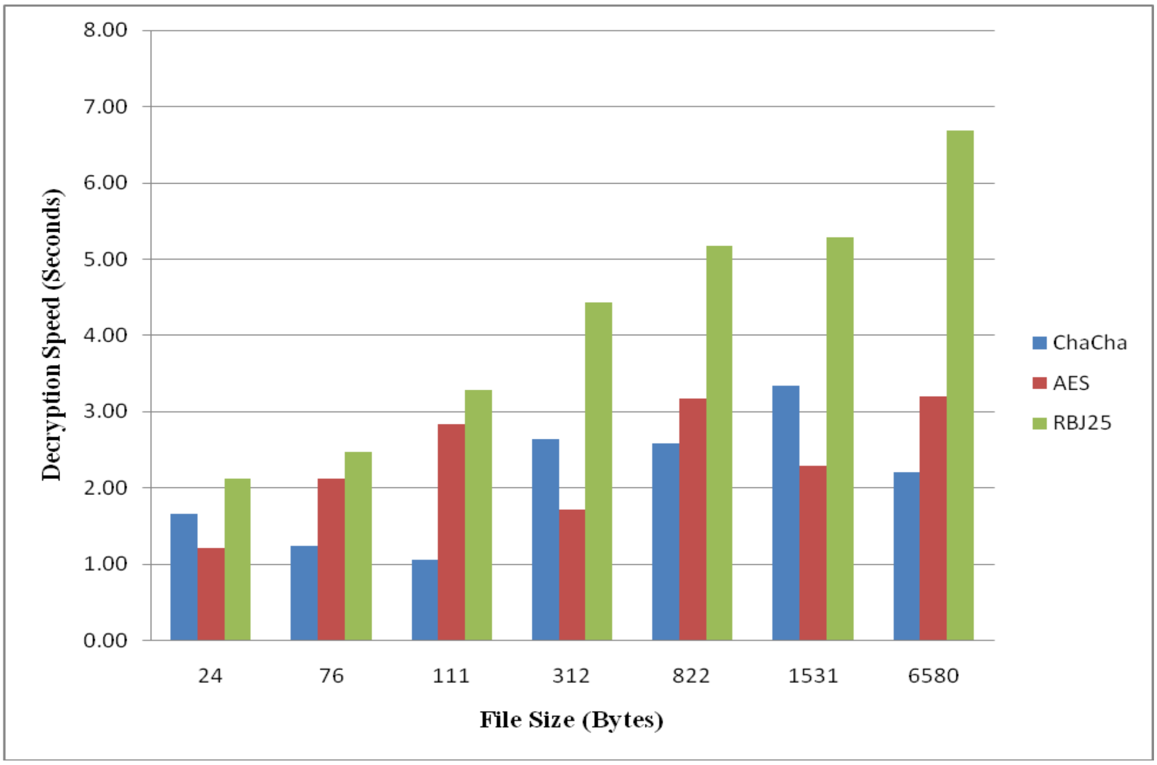

Figure 2. Decryption speed of RBJ25 Algorithm compared with ChaCha and AES

5. Discussion & conclusion

Today’s data need more security is very important to all the areas in the world. For example, bank analysed data, social media analysed data, personal storage data, credit and debit card analysed data, and machine learning algorithm prediction data. To overcome these problem to apply the ChaCha method. This method done only encryption speed with tiny bit data security. So we proposed novel algorithm is RBJ25, it has three part proposed algorithm each layer carrying out some operations in the encryption and decryption part to increase the resistance of the algorithm in such a way that the protection of the data is increased along with effective authentication. Finally the proposed algorithm is implemented using Python and analysed through comparison with traditional AES algorithm and chacha20 security algorithm. In the future, to add the prime factors operations of the data security in the RB26 method for upcoming journals.

References

[1] Dilip Kumar S V, Sikhar Patranabis, Jakub Breier, Debdeep Mukhopadhyay, Shivam Bhasin, Anupam Chattopadhyay and Anubhab Baksi 2017 *A Practical Fault Attack on ARX-like Ciphers with a Case Study on ChaCha20* (Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)) p 33 40

[2] Arun Babu P and Jithin Jose Thomas 2019 *Freestyle, a randomized version of ChaCha for resisting offline brute-force and dictionary attacks* (Journal of Information Security and Applications) arti. 102396.

[3] Alexandre Adomnicai, Jacques Fournier J A and Laurent Masson 2017 *Bricklayer Attack: A Side-Channel Analysis on the ChaCha Quarter Round (*Progress in Cryptology – INDOCRYPT 2017. Lecture Notes in Computer Science, Springer, Cham vol 10698*)* p 65 84

[4] Kazuhide Fukushima, Rui Xu, Shinsaku Kiyomoto and Naofumi Homma 2017 Fault Injection Attack on Salsa20 and ChaCha and a Lightweight Countermeasure (IEEE Trustcom/BigDataSE/ICESS) p 1032 1037

[5] Abdullah Issa, Mohammad A, Al-Ahmad and Abdullah Al-Saleh 2015 *Double-A – A Salsa20 like the Design* (4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT)) p 18-23

[6] Bodhisatwa Mazumdar, Subidh Ali S k and Ozgur Sinanoglu 2016 *A Compact Implementation of Salsa20 and Its Power Analysis Vulnerabilities* (ACM Transactions on Design Automation of Electronic Systems) p 11:1 11:26.

[7] Abdullah Al-Saleh, Mohammed Al-Ahmmad, Abdullah Issa and Adel Al-Foudery 2015 *DOUBLE-A – A Salsa20 like the Security* (4th International Conference on Advanced Computer Science Applications and Technologies) p 24-29

[8] Bodhisatwa Mazumdar, Sk Subidh Ali and Ozgur Sinanoglu 2015 *Power Analysis Attacks on ARX: An Application to Salsa20* (IEEE 21st International On-Line Testing Symposium) p 40-43

[9] Conrad Watt, John Renner, Natalie Popescu, Sunjay Cauligi and Deian Stefan 2019 *CT-Wasm: Type-Driven Secure Cryptography for the Web Ecosystem* (Proceedings of the ACM on Programming Languages) p 77:1- 77:29

[10] Shi Z, Zhang B, Feng D and Wu W 2013 *Improved Key Recovery Attacks on Reduced-Round Salsa20 and ChaCha* (In: Kwon T., Lee MK., Kwon D. (eds) Information Security and Cryptology – ICISC 2012 Lecture Notes in Computer Science, vol 7839) p 337-351

[11] Bagath Basha C and Rajaprakash S 2019 *Securing Twitter Data Using Phase I Methodology* (International Journal of Scientific & Technology Research) p 1952-1955

[12] Bagath Basha C and Rajapraksh 2020 *Enhancing The Security Using SRB18 Method of Embedding Computing* (Microprocessor and Microsystems)

[13] Bagath Basha C and Somasundaram K 2019 *A Comparative Study of Twitter Sentiment Analysis Using Machine Learning Algorithms in Big Data* (International Journal of Recent Technology and Engineering) p 591-599