2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. June 4-5, 2020

Cryptography Encryption Technique Using Circular Bit Rotation in Binary Field

Deepraj Pradhan
Amity Institute of Information Technology,
Amity University,
Noida, India
deeprajpradhan@outlook.com

Subhranil Som
Amity Institute of Information Technology,
Amity University,
Noida, India
ssom@amity.edu

Ajay Rana
Amity Institute of Information Technology,
Amity University,
Noida, India
ajay_rana@amity.edu

Abstract— **In the digital age when technology has advanced so much that communication between people is just a text away no matter where they are geographical. But with every advancement, the technology must be secure enough so that no eavesdropping can take place. That's where cryptography comes into play, Cryptography is necessary when communicating over any untrusted medium. Cryptography is an important field in information security, and, in this modern age, cryptography is used everywhere from surfing the internet to phone calls. The requirement for a better cryptosystem keeps increasing as the advancement of modern computers outpaces the old cryptosystems. In cryptography, the plain text is the original text and ciphertext is the encrypted text. So, the proposed technique in this paper encrypts plain text into ciphertext that is unrecognizable that makes the ciphertext unidentifiable when compared to plain text. First, we generate a very large random number that is at least 1024 bits which will be the key. Encrypting the plaintext is based on the key, first the plain text is divided into blocks of 9 bytes then the key divided into array of 'n/2' number of pairs, the first digit of pair being position in the block and second digit being the number of bits to shift right, the shift is circular from the position of the selected character. In the decryption the left shift is performed to the blocks, the start of decryption begins from the end of the array of key pairs.**

Keywords— ***Bit shift, chi square, binary field, plain text, cryptography***

I. INTRODUCTION

Cryptography is derived from Greek word. It has 2 parts: 'crypto' means "hidden, secret" and 'graphy' means "writing". It is a study of techniques for secure communication in the presence of third parties to maintain information securities such as data integrity, confidentiality, authentication, and non-repudiation. [1] Cryptography manages making reports that can be shared subtly over open correspondence channels. Cryptography is the study of making and utilizing encryption and decryption methods. An encryption calculation works with a key to change the plaintext into ciphertext. Decryption calculation works in the switch request and changes over the ciphertext into plain text. Typically, key is a number that is blended with plaintext to yield ciphertext. The enciphering or encryption is a procedure of changing over plaintext into ciphertext. translating or decryption is a procedure of holding the plaintext from the ciphertext. As a rule, cryptography is utilized to accomplish verification, privacy, uprightness and non-denial to guarantee unwavering quality of information. Cryptosystem is divided into Symmetric Key and Asymmetric Key Cryptography. The field of cryptography is very intriguing and exciting for those who love to hide data. One of the most used cryptosystems is the RSA cryptosystem which utilizes two separate keys, private and public key, it is popular due to the lever of security it provides against any kind of attack even brute force as calculating factors of very large prime number takes several months even for the mightiest computers of modern era.

II. PROPOSED TECHNIQUE

A. Generating Key

The key is generated using random number generators of the size no less than 512 bits i.e. at least 154 digits or 77 pairs of unsigned integers. The key is divided into array of pairs and each pair is generated separately.

Key = [56, 12, 44, 21, …. n]

The first digit of each pair is the location of the character and second digit is the number of bits to shift as show in **Error! Reference source not found.**.

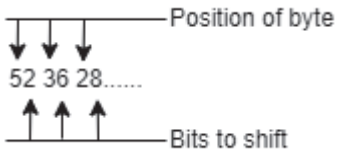

Fig. 1. Sample of key

B. Encryption technique

The encryption technique is done in 3 steps.

1) Divide plaintext into blocks

The plain text is divided into blocks of 10 bytes. The formula to calculate the number of blocks required for text is 'ceiling (length (TEXT)/10)'.

PLAIN_TEXT = "CRYPTOGRAPHY"

BLOCK_1 = "CRYPTOGRAP"

BLOCK_2 = "HY_____"

If the last block does not match the selected block size, then padding "_" can be added to fill the block.

2) Perform a circular right shift rotation

In the encryption technique the bits are shifted to the right and from the end the bits are carried to the byte where the bits were shifted.

The position of bytes ranges from 0 to 9 and bits to shift ranges from 1 to 8. Since a byte consists of 8 bits only, 0 and 9 will invert the bits instead of shifting.

978-1-7281-7016-9/20/$31.00 ©2020 IEEE

Example:

For the purpose of demonstration, only 4 pairs of keys are used.

KEY = [24, 19, 01, 00]

BLOCK_1 = 01000011 01010010 01011001 01010000 01010100 01001111 01000111 01010010 01000001 01010000

BLOCK_2 = 01001000 01011001 01011111 01011111 01011111 01011111 01011111 01011111 01011111 01011111

Rotation 1:

Performing a right shift circular rotation on 3rd byte by 4 bits.

KEY = 24

BLOCK_1 = 01000011 01010010 00000101 10010101 00000101 01000100 11110100 01110101 00100100 00010101

BLOCK_2 = 01001000 01011001 11110101 11110101 11110101 11110101 11110101 11110101 11110101 11110101

Rotation 2:

Here, the number of bits to shift are 9 bits so from 2nd byte to end byte the bits are inverted.

KEY = 19

BLOCK_1 = 01000011 10101101 11111010 01101010 11111010 10111011 00001011 10001010 11011011 11101010

BLOCK_2 = 01001000 10100110 00001010 00001010 00001010 00001010 00001010 00001010 00001010 00001010

Rotation 3:

Performing a right circular shift rotation on 1st byte by 1 bit.

KEY = 01

BLOCK_1 = 00100001 11010110 11111101 00110101 01111101 01011101 10000101 11000101 01101101 11110101

BLOCK_2 = 00100100 01010011 00000101 00000101 00000101 00000101 00000101 00000101 00000101 00000101

Rotation 4:

Here the number of bits to shift are 0 so from 1st byte to end byte the bits are inverted.

KEY = 00

BLOCK_1 = 11011110 00101001 00000010 11001010 10000010 10100010 01111010 00111010 10010010 00001010

BLOCK_2 = 11011011 10101100 11111010 11111010 11111010 11111010 11111010 11111010 11111010 11111010

3) Concatenate all blocks

BLOCK_1 = **"**Þ)Ê, ¢z:' **"**

BLOCK_2 = **"**Û¬úúúúúúúú**"**

CIPHER_TEXT = **"**Þ)Ê, ¢z:' Û¬úúúúúúúú**"**

C. Decryption technique

The decryption technique is done in 3 steps.

1) Divide cipher text into blocks

The cipher text is divided into two blocks of 10 bytes each.

CIPHER_TEXT = **"**Þ)Ê, ¢z:' Û¬úúúúúúúú**"**

BLOCK_1 = **"**Þ)Ê, ¢z:' **"**

BLOCK_2 = **"**Û¬úúúúúúúú**"**

2) Perform circular left shift rotation

In the decryption technique exact opposite of encryption is done that is the bits are shifted to the left and the from the current position the bits are carried to the end position. And like encryption technique the position of a byte ranges from 0 to 9 and bits to shift ranges from 1 to 8. Since a byte consists of 8 bits only, 0 and 9 will invert the bits instead of shifting.

Example:

For the purpose of demonstration only 4 pairs of keys are used.

KEY = [24, 19, 01, 00]

BLOCK_1 = 11011110 00101001 00000010 11001010 10000010 10100010 01111010 00111010 10010010 00001010

BLOCK_2 = 11011011 10101100 11111010 11111010 11111010 11111010 11111010 11111010 11111010 11111010

Rotation 1:

Here the number of bits to shift are 0 so from 1st byte to last byte the bits are inverted.

KEY = 00

BLOCK_1 = 00100001 11010110 11111101 00110101 01111101 01011101 10000101 11000101 01101101 11110101

BLOCK_2 = 00100100 01010011 00000101 00000101 00000101 00000101 00000101 00000101 00000101 00000101

Rotation 2:

Performing a left circular shift rotation on 1st byte by 1 bit.

KEY = 01

BLOCK_1 = 01000011 10101101 11111010 01101010 11111010 10111011 00001011 10001010 11011011 11101010

BLOCK_2 = 01001000 10100110 00001010 00001010 00001010 00001010 00001010 00001010 00001010 00001010

Rotation 3:

Here, the number of bits to shift are 9 bits so from 2nd byte to end byte the bits are inverted.

KEY = 19

BLOCK_1 = 01000011 01010010 00000101 10010101 00000101 01000100 11110100 01110101 00100100 00010101

BLOCK_2 = 01001000 01011001 11110101 11110101 11110101 11110101 11110101 11110101 11110101 11110101

Rotation 4:

Performing a left shift circular rotation on 3rd byte by 4 bits.

KEY = 24

BLOCK_1 = 01000011 01010010 01011001 01010000 01010100 01001111 01000111 01010010 01000001 01010000

BLOCK_2 = 01001000 01011001 01011111 01011111 01011111 01011111 01011111 01011111 01011111 01011111

Concatenate all blocks

BLOCK_1 = "CRYPTOGRAP"

BLOCK_2 = "HY_____"

Remove padding and concatenate.

PLAIN_TEXT = "CRYPTOGRAPHY"

III. PERFORMANCE ANALYSIS

This section analyses the proposed algorithm's time complexity and compares it with AES and RSA encryptions system. AES Encryption is meant for fast encryption and decryption, it could encrypt hundreds of megabytes per second. However, the speed varies with different versions of AES. RSA is public key encryption system i.e. the cryptosystem uses pair of keys for encryption and decryption. RSA has a specific limit to how much it can encrypt in one go; it varies with different key sizes [3].

A. Encryption and decryption comparison

TABLE I. ENCRYPTION COMPARISION WITH .TXT FILES

File Name (.txt)	Size (Bytes)	Encryption Time (Seconds)		
		RSA	AES	Proposed Algo.
file_1.txt	262144	0.828	0.558	4.574
file_2.txt	524288	3.182	0.604	9.291
file_3.txt	786432	7.202	0.625	14.251
file_4.txt	1048576	13.502	0.65	18.959
file_5.txt	1310720	20.09	0.678	23.89

TABLE II. DECRYPTION COMPARISION WITH .TXT FILES

File Name (.txt)	Size (Bytes)	Decryption Time (Seconds)		
		RSA	AES	Proposed Algo.
file_1.txt	262144	0.91	0.359	4.753
file_2.txt	524288	3.948	0.421	9.441
file_3.txt	786432	5.494	0.512	14.518
file_4.txt	1048576	10.469	0.58	19.59
file_5.txt	1310720	16.44	0.624	25.616

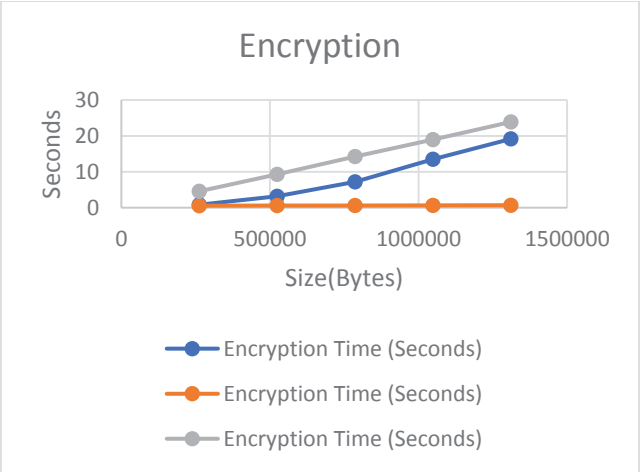

Fig. 2. Encryption chart

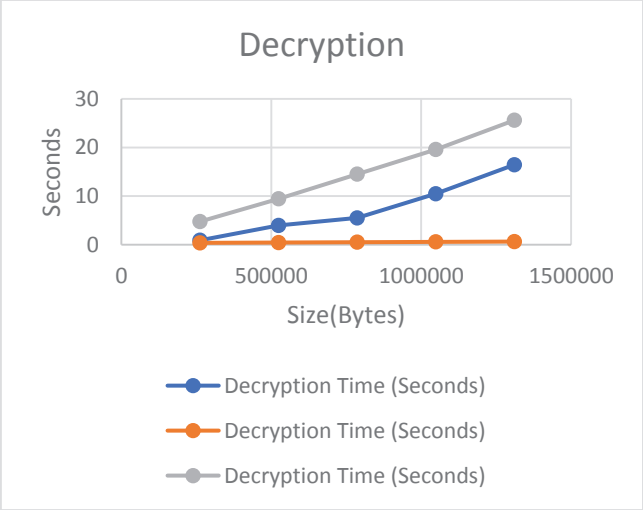

Fig. 3. Decryption chart

B. Chi-Square Test

Chi Square tests have been done to check for non-homogeneity among encrypted files and their source files. Higher Chi Square values shows non-homogeneity of the encrypted files and their source files. Chi Square test are done on text files for Proposed, RSA and AES cryptosystems. The increased value of the Chi Square proves the non-homogeneity for increasing file size. As shown below in Table 3, the higher chi-square values show the non-homogeneity among all three techniques.

TABLE III. CHI-SQUARE TEST

File Name (.txt)	Chi-Square values		
	RSA	AES	Proposed Algo.
file_1.txt	141	141	147
file_2.txt	282	284	295
file_3.txt	424	425	442
file_4.txt	564	567	590
file_5.txt	706	709	736

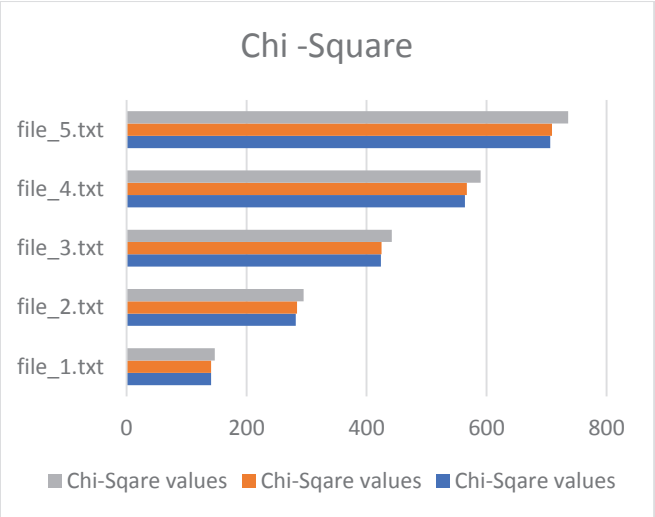

Fig. 4. Chi-Square value chart

IV. CONCLUSIONS

The key point for the proposed technique is that it has similar security as compared to RSA and AES. Even though it has taken longer time in encryption and decryption for large file sizes, the main usability for this encryption would be to encrypt keys for other encryption system or can be used to encrypt small amount of data for cell phones, RFID cards, Smart tags etc. It has a high chi square value which shows that the encryption technique is secure for use.

REFERENCES

[1] A. Gupta and N. K. Walia, "Cryptography Algorithms: A Review," International Journal of Engineering Development and Research 2321-9939, vol. 2, no. 2, 2014.

[2] Davies, Donald. (1997). A brief history of cryptography. Information Security Technical Report. 2. 14-17. 10.1016/S1363-4127(97)81323-4.

[3] N. and W. Wei, "Analysis and Research of the RSA Algorithm," Information Technology Journal, vol. 12, no. 9, pp. 1818-1824, 2013.

[4] S. Som and M. Banerjee, "Cryptographic technique by square matrix and single point crossover on binary field," in 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), Sharjah, United Arab Emirates, 2013.

[5] I. M, G. Devika and K. Shankar, "A Modified Symmetric Key Cryptography Method for Secure Data Transmission," International Journal of Pure and Applied Mathematics, 2017.

[6] Noorunnisa, Nahri & Siddiqui, Rahat. (2016). Review on Honey Encryption Technique. International Journal of Science and Research (IJSR). 5. 1683-1686.

[7] Danasingh, Asir Antony. (2016). Performance Analysis of Data Encryption Algorithms for Secure Data Transmission. International Journal for Science and Advance Research in Technology. 2. 388-390.

[8] Aanjanadevi, S. & Palanisamy, V. & Aanjankumar, s. (2019). An Improved Method for Generating Biometric-Cryptographic System from Face Feature. 1076-1079. 10.1109/ICOEI.2019.8862741.

[9] V. Palanisamy and A. Jeneba mary, "Hybrid cryptography by the Implementation of RSA and AES", International Journal of Current Research, vol. 3, issue . 4, pp. 241-244,April 2011

[10] S. Som, . N. S. Chatergee and J. K. Mandal, "Key based bit level genetic cryptographic technique (KBGCT)," in IEEE, melaka, Malaysia, 2011.

[11] A. Bhardwaj and S. Som, "Study of different cryptographic technique and challenges in future," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Noida, 2016, pp. 208-212.

[12] V. Kunwar, N. Agarwal, A. Rana, J. P. Pandey, " Load balancing in cloud—a systematic review", in Advances in Intelligent Systems and Computing, Vol. 654, pp 583-593 (2018).

[13] B. D. Chauhan, A. Rana, N. Sharma, "Testing sufficiency test (TST) - Evolving a new model for estimating software test cases", in International Journal of Applied Engineering Research, pp 12-21 (2017).

[14] G. Dubey, A. Rana, J. Ranjan, "Fine-grained opinion mining of product review using sentiment and semantic orientation", in International Journal of Business Information Systems, Vol. 25, Issue 1, pp 1-17 (2017).

[15] S. Ghosh, A. Rana, V. Kansal, "Predicting defect of software system" in Advances in Intelligent Systems and Computing, Vol 516 , pp 55-67 (2017).

[16] A. Saroliya, U. Mishra U, A. Rana, "Improvement in routing techniques in P2P networks using a cloud service interface with secure multiparty computation", in Far East Journal of Electronics and Communications, Vol. 16, Issue 3, pp 673-683 (2016).

[17] E. Kashyap, A. Rana, "A Comparative Study of S-shape and Concave Software Reliability Growth Models", in Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015, pp 1452-1455 (2016).

[18] P. Chawla, I. Chana, A. Rana, "Cloud-based automatic test data generation framework", in Journal of Computer and System Sciences, Vol.82, Isuue 5, pp 712-738 (2016).

[19] N. Agarwal N, A. Rana, J. P. Pandey, "Proxy signatures for secured data sharing", in Proceedings of the 2016 6th International Conference - Cloud System and Big Data Engineering, Confluence, pp 255 -258 (2016).

[20] M. K. Shukla, A. Rana, H. Banka, " Classification of the Bangla script document using SVM", in 2016 3rd International Conference on Recent Advances in Information Technology, RAIT 2016, pp 182-185 (2016).

[21] M. Bhardwaj, A. Rana, "Key software metrics and its impact on each other for software development projects", in International Journal of Electrical and Computer Engineering, Vol. 6, Isuue 1, pp 242-248 (2016).