Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 8, No. 3, September 2020, pp.1829-1835

A proposed hybrid cryptography algorithm based on GOST and salsa (20)

Hayder Najm[1,3], Haider K. Hoomod[2], Rehab Hassan[1]
[1]Department of Computer Science, University of Technology
[2] Computer Science Department, Mustanisiryah University, College of Education
[3]Imam Al-kadhum College (IKC), Computer Technique Engineering Department

ABSTRACT

Security concepts are frequently used interchangeably. These concepts are interrelated and share similar objectives for the protection of privacy, credibility, and access to information; however, there are some slight differences between them. Such variations lie mostly in the subject matter approach, the approaches used, and the focus fields. With the intention of protecting data in contradiction of unauthorized or unintentional disclosure, cryptography is used during transit (electronic or physical) and when data is stored. In the course of the past few years, some block ciphers and stream ciphers have been proposed. These block ciphers take encryption method that uses Substitution-Permutation and Feistel network structure while stream ciphers choose a onetime method. GOST encryption is based on the confidentiality of the secret key. However, it leads to the same ciphertext being generated when the encryption program is used with the same key for the plain text. Reproduction of messages can thus easily be identified by an opponent that is a weak link in any communication. In this paper, proposed a hybrid encryption method based on GOST block cipher and Salsa stream cipher to provide proper security with as high hardness randomly enhances the five standard tests and modifies key schedule as secure operations. The downside of the GOST algorithm is a simple key schedule so that in certain circumstances be the weak point of the method of cryptanalysis as related-key cryptanalysis. However, this resolved by the proposed method by passing the keys of GOST to Salsa stream to have the right combination and more robustness security. Its need for 2^{256} probable keys to breaking keys that, because of its uncomfortable procedure in this situation, is to be not used brute force attack. Correspondingly, five standard tests successfully surpassed the randomness of a proposed method.

Keywords: Information Security; Hybrid Cryptography; Stream Cipher; Block Cipher; GOST; Salsa(20)

Corresponding Author:

Hayder Najm Abd Al Sada,
Department of Computer Science
University of Technology, Baghdad, Iraq
E-mail: cs.19.87@grad.uotechnology.edu.iq

1. Introduction

Information security means defense against unauthorized access, use, secrecy, interrupt, alteration, or degradation of the information and information systems. All of the security concepts of security are often used interchangeably. These areas are interrelated and share similar objectives for the protection of privacy, credibility, and access to information; however, there are some slight differences between them. Such variations lie mostly in the subject matter approach, the approaches used, and the focus fields [1].

In order to protect information against unauthorized or unintentional disclosure, cryptography is used during transit (electronic or physical) and when data is stored. The method of translating common knowledge (plain text) in unintelligible gibberish (ciphertext) applies almost exclusively to encryption. Decryption is the opposite, turning the ciphertext type into plaintext [2]. The cipher is a matched pair of methodologies that create encryption and reverse decryption, and the comprehensive operation of the cipher is controlled by the algorithmically and the key in each specific instance. Keys are critical, as ciphers are trivial breakable without variable keys and thus typically less than useful [3], as shown in Figure 1.

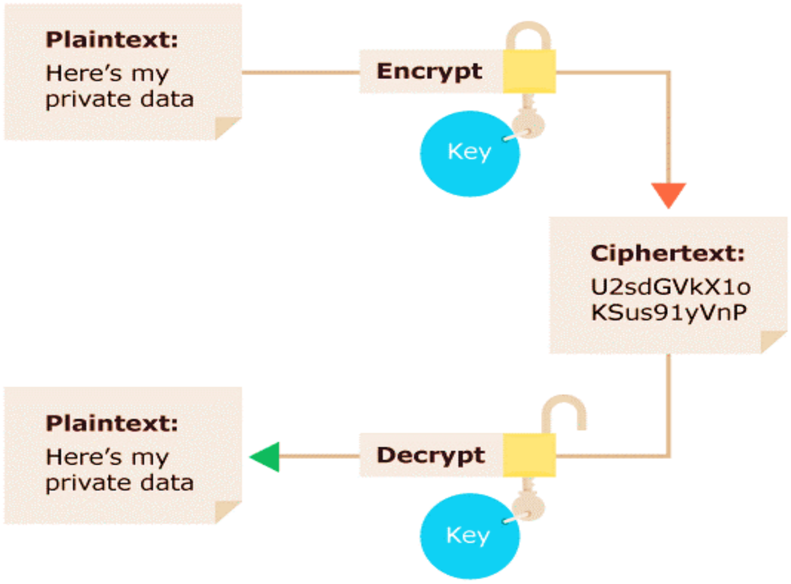


Figure 1. Operations for encrypting and decrypting [3]

Symmetric cryptography consists of taking and encrypting a piece of data and a key so that the same key will be used to decipher the data. It is a widespread type of encryption [4]. Ciphering methods are often classified into two kinds: the first kind is a block cipher that means the technique of the cipher by cutting each actual data into consecutive blocks, and each block is ciphering using the same key, as shown in Figure 2 [5].

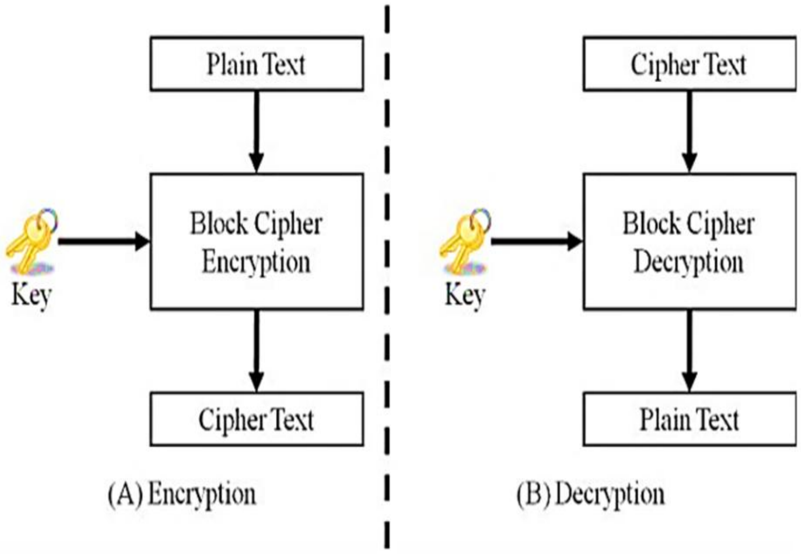


Figure 2. Cryptography algorithm (block cipher): a – Encryption; b – Decryption [6]

The block consists of coupled methodologies, first for encrypted data, E, and the second for decrypted data, E-1. Both methodologies two inputs agree simply, with block size n bits and k bits for a key. The inverse function of encoding standards for all fixed keys is the decryption function, so[6]:

$$E^{-1}_{k}(E_{k}(M))$$

The second kind is a stream cipher that means the cipher technique by using the XOR function between the actual data and the randomized cipher sequence. The cipher and decode techniques of the stream cipher can be seen in the below equations [7]:

$$C[s]=O[s] \oplus K[s] \qquad (1)$$

$$O[s]=C[s] \oplus K[s] \qquad (2)$$

C[s] means the data bit cipher, K[s] means the random sequence bit key, O[s] means the actual data bits, and S is 1 bit at that same time. Going to focus on the same equation (1) and the same (2), the cipher and decode combined demanded that the same seed key used to generate the identical keystream sequence K[s] as seen in Figure 3 [8]. The modification of the keystream sequence would not require an adversary's guide to split the cipher data by offering a keystream style that could not be recurring [9].

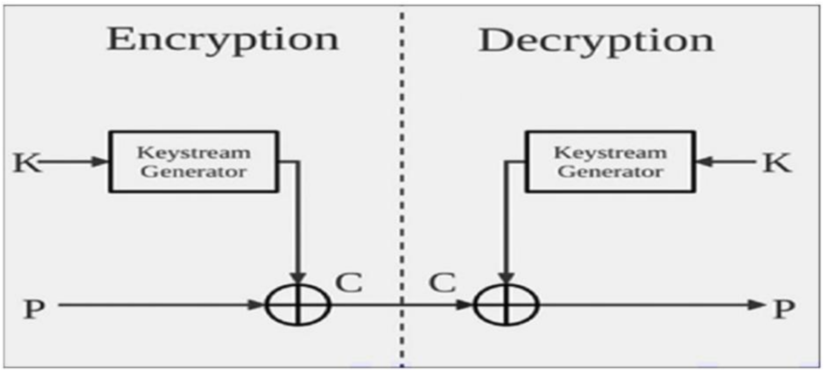


Figure3. Stream cipher [8]

1.1. GOST

GOST stands for "Gosudarstvennyi Standard" or "Government Standard.”. Throughout the Cold War, the Soviet Union developed this method of covering sensitive information during communication [10]. It is a simple algorithm for encryption that uses a 64bit block with the 256 bit key, with processes up to 32 rounds. The GOST approach also uses the various S-Box 8 components, the Circular Shift Left and XOR operations [11]. The weakness known until now is that its key schedule is simply that, under certain circumstances, it is the weak point of the Cryptanalysis Method as Related-Key Cryptanalysis. It can, however, be solved by passing the keys onto a powerful cryptographic hash function, for example, SHA-1, and then using the results to enter an initialization hash key. The benefit of this approach is the GOST speed very well, but not so fast as IDEA Blowfish [12].

Structure of the GOST [13]

1. A 256-bit string Key Store Unit (KSU) can be saved by 32-bit registration (Key0, Key1... Key7).
2. Two registers of 32-bits (R1, R2)
3. Modulo 232 (CM1), 32-bit adder
4. Add XOR (CM2) bit by bit
5. Sub Block (S), 64-bit SBox eight.
6. Register of the left rotation shift (R),11 bit.

The key structure process is the mechanism by which the password is formulated to encrypt the plaintext. It can be presented as follows [14]:

1. Input key, 256-bit key (key1, key2, key3, key4… key256)
2. Generating of 8 KSU

Key0 = (key32… key1)
Key1 = (key64… key33)
Key2 = (key96… key65)
Key3 = (key128… key97)
Key4 = (key160… key129)
Key5 = (key192… key161)
Key6 = (key224… key193)
Key7 = (key256… key225)

1.2. Salsa (20)

Salsa (20) is the counter mode used for an encryption stream cipher. As shown in Figure 4, Salsa's original seed (20) is an array (4, 4) of 512 bits [15].

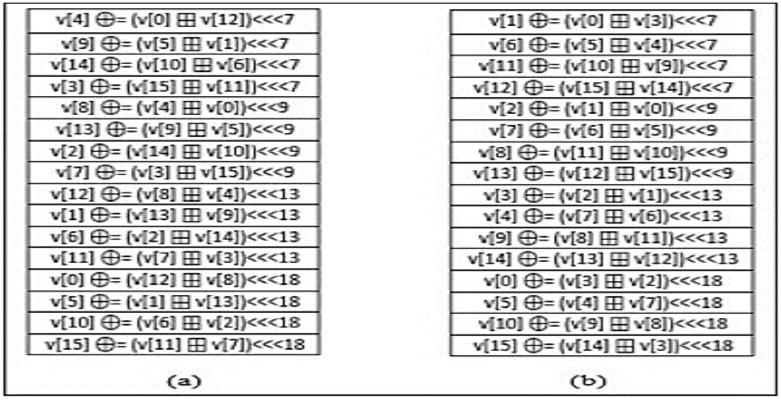
Constant 1	Key1	Key2	Key3
Key4	Constant 2	Nonce1	Nonce2
Counter1	Counter2	Constant 3	Key5
Key6	Key7	Key8	Constant 4

=

V0	V1	V2	V3
V4	V5	V6	V7
V8	V9	V10	V11
V12	V13	V14	V15

Figure 4. Distribution of Salsa (20) Array [15]

The primary operations at Salsa (20), as shown in Figure 5, are "addition, XOR and rotation" and applied for 10 rounds in the Salsa(20) array. Each round, the Salsa(20) Array is changed twice, so it's called Salsa (20). At the end of the Salsa (20), the additional operation is used between the final adjustment of the Salsa (20) Array and the initial seed Salsa (20) Array.



v[4] ⊕= (v[0] ⊞ v[12])<<<7
v[9] ⊕= (v[5] ⊞ v[1])<<<7
v[14] ⊕= (v[10] ⊞ v[6])<<<7
v[3] ⊕= (v[15] ⊞ v[11])<<<7
v[8] ⊕= (v[4] ⊞ v[0])<<<9
v[13] ⊕= (v[9] ⊞ v[5])<<<9
v[2] ⊕= (v[14] ⊞ v[10])<<<9
v[7] ⊕= (v[3] ⊞ v[15])<<<9
v[12] ⊕= (v[8] ⊞ v[4])<<<13
v[1] ⊕= (v[13] ⊞ v[9])<<<13
v[6] ⊕= (v[2] ⊞ v[14])<<<13
v[11] ⊕= (v[7] ⊞ v[3])<<<13
v[0] ⊕= (v[12] ⊞ v[8])<<<18
v[5] ⊕= (v[1] ⊞ v[13])<<<18
v[10] ⊕= (v[6] ⊞ v[2])<<<18
v[15] ⊕= (v[11] ⊞ v[7])<<<18

(a)

v[1] ⊕= (v[0] ⊞ v[3])<<<7
v[6] ⊕= (v[5] ⊞ v[4])<<<7
v[11] ⊕= (v[10] ⊞ v[9])<<<7
v[12] ⊕= (v[15] ⊞ v[14])<<<7
v[2] ⊕= (v[1] ⊞ v[0])<<<9
v[7] ⊕= (v[6] ⊞ v[5])<<<9
v[8] ⊕= (v[11] ⊞ v[10])<<<9
v[13] ⊕= (v[12] ⊞ v[15])<<<9
v[3] ⊕= (v[2] ⊞ v[1])<<<13
v[4] ⊕= (v[7] ⊞ v[6])<<<13
v[9] ⊕= (v[8] ⊞ v[11])<<<13
v[14] ⊕= (v[13] ⊞ v[12])<<<13
v[0] ⊕= (v[3] ⊞ v[2])<<<18
v[5] ⊕= (v[4] ⊞ v[7])<<<18
v[10] ⊕= (v[9] ⊞ v[8])<<<18
v[15] ⊕= (v[14] ⊞ v[3])<<<18

(b)

Figure 5. Salsa (20) Operations: a. Changing the first; b. Changing the second [16]

In each round of Salsa (20) ninety-six-word operation has been completed, i.e., Forty-eight-word action for first change followed by forty-eight-word action for the second change. Forty-eight words operation is determined by multiplying 16-word operation (addition, XOR, and rotation) by three operations. For ten rounds, nine hundred sixty word operations are the number of operations. At the end of Salsa [20], 960-word operations plus 16-word operations conclude a total of one hundred and 72-word operations for one encryption [16].

2. Related work

Many different papers are talking about GOST and Salsa (20) individually. In [17], The first outcome of the GOST attack with reduced rounds number, using the differential form, was represented. The average use of 251 -plaintexts selected key 13-round GOST could be obtained for S-Boxes used as a submission to the Russian Federation Central Bank. When keys make the difference chance, the largest, 17-round GOST can be targeted. The research is often broadened by the combination of a key-assault. The main of 21-round GOST can be obtained from 256choice plaintexts. In [18], show that GOST is not stable against (advanced) differential cryptanalysis (DC), Russian researchers have previously postulated that GOST would be safe against DC for only seven rounds out of 32. Japanese researchers have already broken about 13 rounds. In [19] a modern process of Salsa (20) was developed by quicker diffusion according to the Chaos theory than the original Salsa(20). The majority of experience shows that a modern two-iteration technique is faster than the four simple iterations, although it has the same degree of diffusion. In, [20, 21] the 512 bit Salsa (20) array (4, 4) has been altered to 576 bits Salsa (20), i.e., Each location in the Salsa (20) sequence is used with 64-bit words and changing its position by nine operations in each iteration led to more diffusion than the standard Salsa (20). Each location is used. Many papers used to divide Salsa (20).In [22], the algorithm relies on certain elements to choose the method of encryption (logical operation (XOR, AND) between a secret key and the plain text by encryption and decryption. Select from the main essential elements of intelligence. The random generator secret key is encrypted and sent in encrypted text via the RSA algorithm. In [23], the work aims to tackle all key (block and stream) symmetrical encryption algorithms, irrespective of the encryption algorithm, to produce high randomness, and different lengths, so it is suitable for all algorithms needed by the encryption agency.

3. Proposed system

The main idea of the proposed system is used Gost algorithm and salsa stream as a combination of a block cipher and stream cipher by combing the benefit of each other. Key schedule of Gost algorithm will not be used, the keys generated by salsa stream will be utilized as keys to Gost algorithm to handle the weak point in GOST algorithm. So, the generated keys of Salsa will be generated keys by 20 rounds to mix keys and present robustness keys. The plain text will be encrypted into 32 rounds, as shown in Figure 6 by used the various keys of the salsa stream in each round. The decryption process is the same encryption process but in the reverse key of salsa stream.

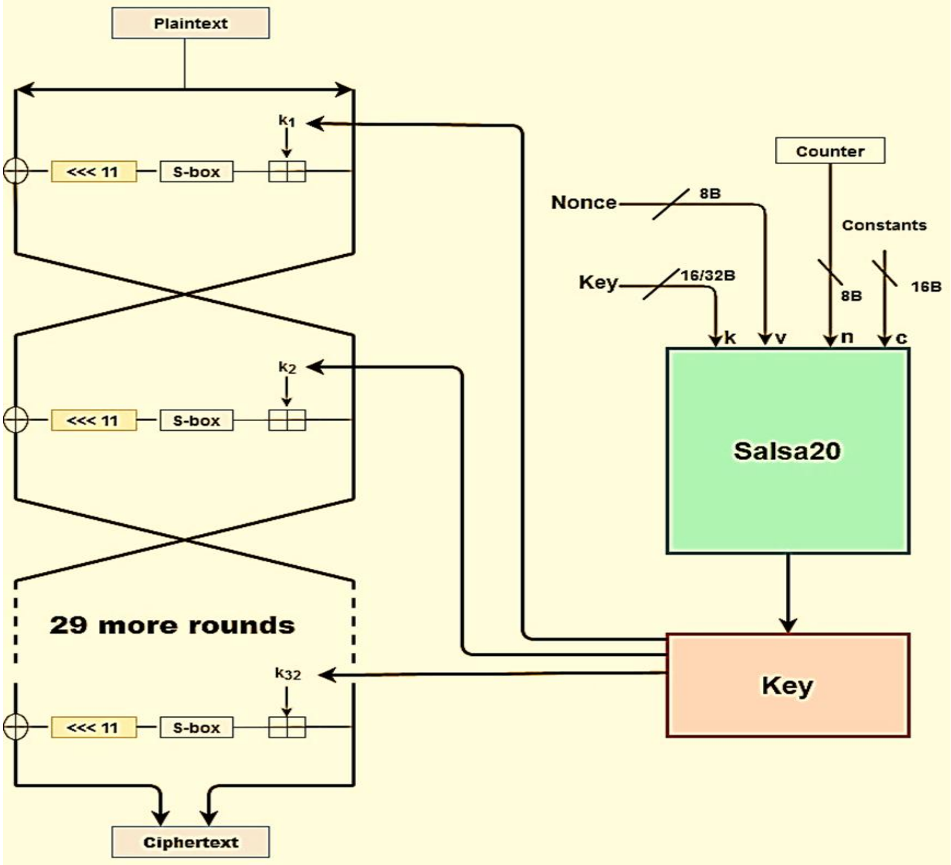


Figure 6. The proposed system block diagram

This encryption of texts is based on the various keys of the salsa stream in each round. The decryption process is the same encryption process but in the reverse key of salsa stream.

4. Result

Hybrid cipher, block and stream cipher, are the glorious fusion. Hybrid is a cryptographic procedure which has both block cipher, and stream cipher characteristics, that ensure the encryption and decryption scheme in both software and hardware is straightforward and easy to implement for resource-controlled devices like smart devices and wireless nodes. This work examines the possibility of mixing a block cipher and a stream cipher to produce a strong hybrid cipher. It involves a specific proposal for mixing GOST 256-bit and Salsa 256-bit keys cipher that is considerably faster than the traditional GOST and probably more secure. The encryption process requires input plaintext data 64-bit or 16 hex digits or eight characters through 32 iteration stages (rounds), while the decryption process involves reverse of the encryption process. The power consumption is lower, and the encryption speed in the system is more rapid. Our proposal is designed to be used in high security required low-cost devices as it is resistant to most of the cryptanalytic attacks common to block ciphers and stream ciphers. According to a table 1, the proposed system delivers good security and reliability.

Table 1. Five standard tests

Test	Value	Threshold
Frequency	1.231	3.481
Serial	2.124	5.991
Poker	14.912	24.995
Run	7.582	12.591
Auto-correlation	0,814	1.96

According to Table 1, the proposed system delivers good security and reliability.

5. Conclusion

The downside of the GOST algorithm is a simple key schedule so that in certain circumstances is the weak point of the method of cryptanalysis as related-key cryptanalysis. However, this resolved by the proposed method by passing the keys of Gost to Salsa stream to have the right combination and more robustness security. Its need for 2^{256} probable keys to breaking keys that, because of its uncomfortable procedure in this situation, is to be not used brute force attack. Also, five standard tests successfully surpassed the randomness of a proposed method.

6. References

[1] W. Mao, Modern cryptography: theory and practice. Pearson Education India, 2003.

[2] M. Y. Rhee, Internet security: cryptographic principles, algorithms and protocols. John Wiley & Sons, 2003.

[3] A. G. Konheim, Computer security and cryptography. John Wiley & Sons, 2007.

[4] S. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo, “Security Analysis of a Cryptographically-Enabled RFID Device.,” in USENIX Security Symposium, 2005, vol. 31, pp. 1–16.

[5] T. W. Cusick, C. Ding, and A. R. Renvall, Stream ciphers and number theory. Elsevier, 2004.

[6] E. Bach, J. O. Shallit, J. Shallit, and S. Jeffrey, Algorithmic number theory: Efficient algorithms, vol. 1. MIT press, 1996.

[7] M. Hell, T. Johansson, A. Maximov, and W. Meier, “A stream cipher proposal: Grain-128,” in 2006 IEEE International Symposium on Information Theory, 2006, pp. 1614–1618.

[8] Y. Minglin and M. Junshuang, “Stream ciphers on wireless sensor networks,” in 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 2011, vol. 3, pp. 358–361.

[9] B. Schneier, Applied cryptography: protocols, algorithms, and source code in C. john wiley & sons, 2007.

[10] L. Babenko, “Algebraic cryptanalysis of gost encryption algorithm,” J. Comput. Commun., vol. 2, no. 04, p. 10, 2014.

[11] N. T. Courtois, “Cryptanalysis of gost in the multiple-key scenario,” Tatra Mt. Math. Publ., vol. 57, no. 1, pp. 45–63, 2013.

[12] M. Iqbal, Y. Sahputra, and A. P. Utama Siahaan, “The Understanding of GOST Crytography Technique,” Int. J. Eng. Trends Technol., vol. 39, no. 3, pp. 168–172, 2016, doi: 10.14445/22315381/ijett-v39p229.

[13] A. A. Dmukh, D. M. Dygin, and G. B. Marshalko, “A lightweight-friendly modification of GOST block cipher,” Математические вопросы криптографии, vol. 5, no. 2, pp. 47–55, 2014.

[14] N. T. Courtois, “An improved differential attack on full GOST,” The new codebreakers, Springer, 2016, pp. 282–303.

[15] D. J. Bernstein, “The Salsa20 Family of Stream Ciphers, New Stream Cipher Designs: The eSTREAM Finalists.” Springer-Verlag, Berlin, Heidelberg, 2008.

[16] D. Priemuth-Schmid and A. Biryukov, “Slid pairs in Salsa20 and Trivium,” in International Conference on Cryptology in India, 2008, pp. 1–14.

[17] H. Seki and T. Kaneko, “Differential cryptanalysis of reduced rounds of GOST,” in International Workshop on Selected Areas in Cryptography, 2000, pp. 315–323.

[18] N. Courtois and M. Misztal, “Differential Cryptanalysis of GOST.,” IACR Cryptol. ePrint Arch., vol. 2011, p. 312, 2011.

[19] M. Almazrooie, A. Samsudin, and M. M. Singh, “Improving the Diffusion of the Stream Cipher Salsa20 by Employing a Chaotic Logistic Map.,” JiPS, vol. 11, no. 2, p. 310, 2015.

[20] A. Issa, M. A. Al-Ahmad, and A. Al-Saleh, “Double-A--A Salsa20 Like: The Design,” in 2015 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT), 2015, pp. 18–23.

[21] M. Salih Mahdi and N. Flaih Hassan, “a Suggested Super Salsa Stream Cipher,” Iraqi J. Comput. Informatics, vol. 44, no. 2, pp. 1–6, 2018, doi: 10.25195/2017/4422.

[22] A. K. Farhan, “Proposed Hybrid Approach of Stream Cipher Base on Selector of Encryption operation and Key Symmetric Translate,” vol. 29, no. 11, 2011.

[23] F. T. Abd El Hussien, “Proposed Algorithm To Generate Encryption Key For Block And Stream Cipher Using DNA Computing,” Iraqi Journal of Information Technology, vol.8, no. 3, pp. 68-82, 2018, doi: 10.34279/0923-008-003-008.