International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 06, 2020
ISSN: 1475-7192

Hybrid Method for Encoding of Genetic and RC4 Algorithms

[1]Marwah Kamil Hussein, [2]Kasim Al-Salim, [3]Asaad Alhijaj

ABSTRACT--*In this paper, the proposed new method used is to generate the encryption key used in the RC4 algorithm by using the genetic algorithm by randomly generating it using the Rand function, and then subjecting it to the randomized conditions approved. If a match is used for coding, otherwise the genetic algorithm will be used to generate the random genetic key, which is along the length of the text to be encrypted.The proposed method used a new structure to hide the encrypted key within the transferred text, in addition to the integrity of the transferred key (integrity) was confirmed by using a simple flux function.*

Keywords-- *Coding, RC4, Genetic Algorithms, Voice, Hash Algorithm.*

I. INTRODUCTION

Information security at the world level is an obsession and concern for those in charge of managing various information systems, especially in light of the growing information crime operations that imposed the need for concerted efforts by all countries and governmental and private institutions worldwide, including individuals, to eliminate all violations of information security In particular, in light of the development of technology and the spread of risks involved in its various uses and applications [1].

The term information security is defined in its broad and comprehensive concept as a set of procedures that enables the owner of the information to keep his personal information, data and financial and bank accounts under his full and direct control, and not to allow any unauthorized person to access it with a view to circulating it either in good faith or in bad faith or with the aim of blackmail and tampering It out. Security dysfunctions in information security systems usually occur when the systems are compromised through, for example, hackers, viruses, or any other type of malicious program. Information security professionals strive to ensure the integrity of the various information systems by achieving three basic requirements, which are confidentiality of information, integrity of information, and availability of information [2].

- The fulfillment of the first requirement related to confidentiality of information requires that information be secured in a way that only authorized persons can access it (others with authority or authorized).
- As for the second requirement of confidentiality of information, it seeks to ensure the integrity of the sources of information, so that it cannot be changed or updated only by authorized persons only.
- The third requirement to ensure information security is the possibility and ease of providing information when it is needed. There are also a number of elements that weaken the security systems of different networks and

[1]*University of Basra, College of Computer Science and Information Technology, Computer Information Systems Dept, Basra, Iraq, lava85k@gmail.com*
[2]*University of Basra, College of Computer Science and Information Technology, Computer Dep., Basra, Iraq
Kasim, alsalim@gmail.com*
[3] *University of Basra, College of Computer Science and Information Technology, Computer Information Systems Dept. Basra, Iraq , Zenhjaj@yahoo.com*

Received: 22 Feb 2020 | Revised: 13 Mar 2020 | Accepted: 05 Apr 2020

steal information, among which are, for example, password leaks, electronic eavesdropping, hacking, viruses, lures, and identity theft [3][4].

From this standpoint, we know the extreme importance that necessitated the attention of states, governmental and private institutions and individuals, and in light of the development of technology and the spread of risks that have become a problem for societies, technology despite its great services to the human being in the modern era, but it is like any other invention that has many advantages and has serious drawbacks if it does not identify It has to work to avoid it, and in the context of our research we discussed methods and tools to protect the information security of the average user in using the RC4 encryption algorithm [5].

RC4 was designed by Ron Rivest of RSA Security in 1987. While it is officially termed "Rivest Cipher 4", the RC acronym is alternatively understood to stand for "Ron's Code"[6] . The RC4 name is trademark, so RC4 is often referred to as ARCFOUR or ARC4 (this means the alleged RC4) [12] to avoid trademark problems [7].

Noting these advantages, the GA algorithm was used in this research to generate the encryption key used in the RC4 algorithm. After reviewing some previous research that dealt with the uses of GA with coding and decoding, for example:

Where researcher [8] has used the genetic algorithm to find the best key to use for encoding texts in a compensatory coding method (Substitution cipher). As for the researcher [9], GA was used to find the length of the secret key that will be used in the analysis of the Permutation cipher. In addition, the search used [10] GA to break the cipher transposition encoded text, and finally researchers [11] introduced three methods of guesswork intuition to reach optimization: (used in the simulation (annealing, genetic algorithm), which was used in the used Transposition cipher [12].

In this research, the genetic algorithm was utilized by proposing a new method for generating the random key used in the RC4 algorithm.

II. STREAM CIPHER OF RC4 ALGORITHM

- Stream cipher operate on a stream of data, one byte at a time.
- Typically stream ciphers perform an Exclusive OR(XOR) operation on a stream of plaintext bytes with the key stream from a pseudo random number generator (PRGA).
- Decryption is achieved by the same byte wise XOR operation on the cipher text.
- Fast and easy to implement in hardware.
- PRGA guidelines:-

- The key stream must have a large period making the repetition of the a sequence for a part.
- The key stream generated should possess as much properties as a true random number generator.
- The input master key (K) must be as a large as possible. [13][14].

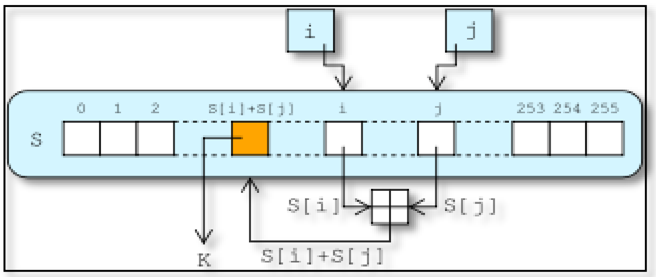

Figure 1: PRGA flowchart.

III. SECURE HASH ALGORITHM (SHA)

Is one of the Hash algorithms that find a constant string of any text or file, there are several types, including:

SHA1: This algorithm produces a 160 bit (20 byte) key.

SHA512: This algorithm produces a 512 bit (64 byte) key of any message or file of any length and a number of other SHA types as we see it in Figure 4 below and the property of each type. We'll explain in detail how SHA512 works [15][16].

3.1 Part (1):

We know that SHA512 receives any length of data and finds it has a 512 bit HASH length in the Fig. 2. It divides the message into a block each one has a size of 1024 bit and another 128 bit in the last block it is reserved for the length of the real data i.e. this algorithm can find HASH for data with a maximum length of its length 2 ^ 128 and the bits that remain blank between the last 128 bit and the actual data of the message after converting it to binary, we have padding, meaning we enter one number and a number of zeros follow it until we fill in the empty bits. And the last block accepts only 896 bits because as we said the last 128 bits in the last block are reserved for the length of the real message in binary system format where iv = H0 represents the eight registers (A, B, C, D, E, F, G, H) each one of its size 64 bit total is 512 bit which will eventually represent the message's Hash. These are initial values stored within Registers. See Fig. 2 [17].

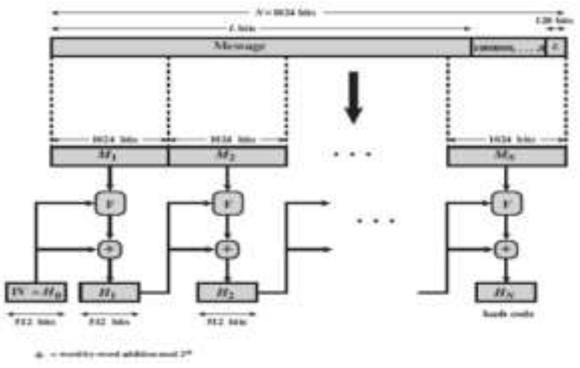

Figure 2: Message digest generation using (SHA-512).

3.2 Part (2)

Which represents Part 2 in the Fig. 2 is the letter (F) in Part No. 1, where it repeats its same operations with each block so we will explain on one Block and the rest of the same thing. Fig.3 represents the operations that will be performed on each block of the real message to produce a key whose length is 512 bits stored in (A, B, C, D, E, F, G, H) and is considered as an entry for operations on the next Block if the data is more than Block As shown in Fig. 3, or the final result is considered if the data is a single block. And that each (F) is divided into 80 Round, each one carrying out the operations inside him once (Part 4 in Fig. 5 represents the operations that will take place within each Round) . In the Fig. 3, each round of 80 is entered with a value of a certain K between (K0-K79), which are fixed values consisting of 64 bits taken from the following table [17].

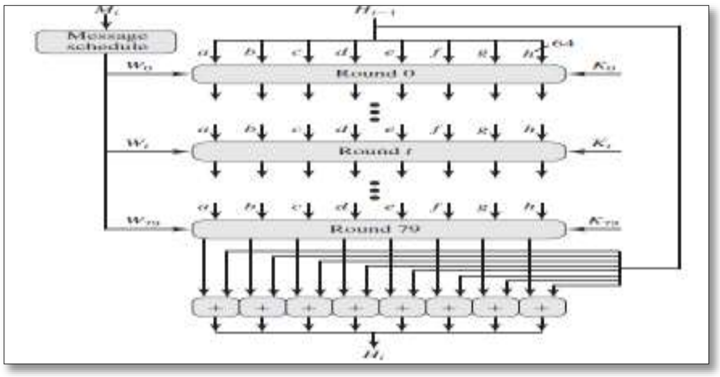

Figure 3: SHA-512 processing of a Single 1024- bit block.

3.3 Part (3):

We also note that in the Fig. 4, each Round enters a value from Block data of 64 bit length and that the length of one Block is 1024 bit, so it divides (1024) into 16 Block, each one of which is 64 bit size represented (W0-W15) and enters the first 16 Round and the rest of Round takes Wt according to the following formula [18].

$W_t = \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16}$

where

$\sigma_0^{512}(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x)$

$\sigma_1^{512}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x)$

$ROTR^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

$SHR^n(x)$ = left shift of the 64-bit argument x by n bits with padding by zeros on the right.

+ = addition modulo 2^{64}

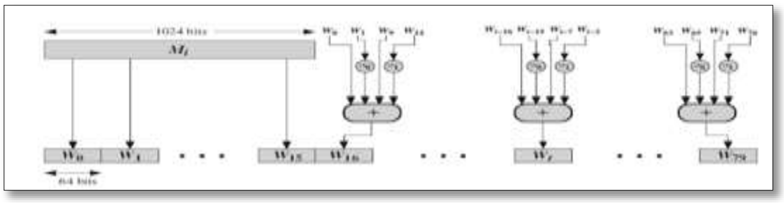

Figure 4: Creation of 80-word input Sequence for SHA-512 processing of single block.

3.4 Part (4):

Part No. 4 in Fig. 5 represents the operations that will take place within each Round in order to not update them on the Registers values that will consider the updated values as the next Round entry and in the last Round is the Hash of the message if the message is from one block and otherwise it is considered an entry for Registers data in the next Block W_t Block: represents 64 bit of Block data previously explained in preparation for each Round [18]. K_t: These are 64-bit constant values taken from the table previously explained

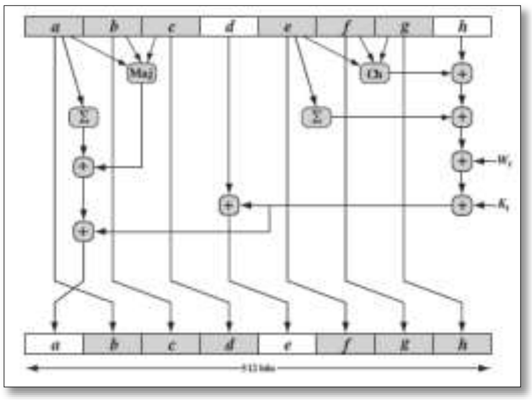

Figure 5: Elementary sha-512 operation (single round).

$$T_2 = (\textstyle\sum_0^{512} a) + Maj(a,b,c)$$

$h = g$, $g = f$, $f = e$, $e = d + T_1$, $d = c$, $c = b$, $b = a$, $a = T_1 + T_2$

where

t = step number, $0 \leq t \leq 79$

$$Ch\ (e,f,g) = (e\ AND\ f) \oplus (NOT\ e\ AND\ g)$$

The conditional function: If e then f else g

$$Maj\ (a,b,c) = (a\ AND\ b) \oplus (a\ AND\ c) \oplus (b\ AND\ c)$$

The function is true only of the majority (two or three) of the arguments are true

$$(\textstyle\sum_0^{512} a) = ROTR^{28}(a) \oplus ROTR^{34}(a) \oplus ROTR^{39}(a)$$

$$(\textstyle\sum_1^{512} a) = ROTR^{14}(e) \oplus ROTR^{18}(e) \oplus ROTR^{41}(e)$$

$ROTR^n(x)$ = circular right shift (rotation) of the 64-bit argument x by n bits

W_t = a 64-bit word derived from the current 512-bit input block

K_t = a 64-bit additive constant

IV. GENERAL STEPS OF THE GENETIC ALGORITHM

- Create a primary generation
- Find the function of the objective, the extent of fitness, and the potential contribution to the primary generation
- Sections
- Parent test (Selection).
- Interventional intervention (Crossover)
- Change within one syllable (Mutation)
- Standard for stopping the genetic algorithm [19][20]

V. RESEARCH OBJECTIVE

To obtain more confidentiality, faster implementation and less cost, an algorithm was proposed. Hybrids take advantage of the genetic characteristics to make a key and check its randomness by using approved methods. Instead of regenerating the key at the receiving end, a new method has been used to hide the key. Inside the encoded text before sending, the integrity of the received key has been confirmed by using the SHA camouflage function.

VI. SUGGESTED WAY TO CONFIGURE THE KEY

Genetic algorithm has been relied upon to avoid behaviors caused by the traditional method. In Initially, a random generation of key is generated using the rand function in Matlab, noting that the length of the key must be the length of the text to be encoded, and then the tribe is measured from the randomness**.** As per the approved conditions [5], if the conditions are fulfilled, the individual is random and will be used for coding**.** The whole primary generation does not fulfill the conditions the genetic algorithm is used.

VII. STRUCTURE OF THE PROPOSED METHOD

The basis for successful encryption is the approved algorithm and key confidentiality. The following algorithm has been adopted:

- Algorithm *(1)*

1. Getting Started

2. Enter the express text to be encrypted

3. Convert the express text to the binary code using the ASCII code. 4. Calculate the number of bits in which the text is made after converting it to the binary system. 5. The code is shown in the code with the code.

4. Random Key Test According to Algorithm (2)

5. In the event that it does not fulfill the conditions of the algorithm (2), the genetic algorithm (3) is used.

6. If the conditions are met, the key is used to encode the express text using the RC4 encoding method.

7. SHA Flux Calculation for the key and result set at end of text after encoding

8. Hide the key within ciphertext according to algorithm (4(

9. Calculate the number of characters of the original text and place it at the end of the text

10. Send the message

11. End.

- Algorithm *(2).*

The approved conditions were used to randomize the test [5] as follows:

1. Getting Started
2. F = 0
3. If the bilateral ranks are equal to "1" = the bilateral ranks are equal to "0", then F1 = 1, otherwise F1 = 0.
4. If there is a block of bilateral orders measured by n and there is no gap between the two orders of size, then F2 = 1 or F2 = 0.
5. If there is a gap of the two orders of magnitude n-1 then F3 = 1 or F3 = 0

6. Calculate the function F = F1 + F2 + F3
7. If the result is F = 3, the individual is random, and otherwise the individual is not random, the genetic algorithm will be used.
8. The End.

- Algorithm (3)

1. Getting Started
2. Generation of a random generation called the primary generation
3. Calculate the function of F = Fitness and its order
4. Find the probability by dividing by the value (Fitness / Fitnesses sum)
5. Perform the Selection process using a roulette wheel, by randomly generating new ones as well.
6. Then, crossover process between the new and old generation
7. Mutation randomly performed on the new generation
8. A percentage of the old and new generation is chosen according to 40 old to 60 new.
9. Randomization process of randomization test again on the new generation.
10. The End

- Algorithm (4)

A new algorithm has been proposed to hide as follows:

1. Getting Started
2. The switch is converted to ASCII coding
3. It is enclosed within a randomly encoded text
4. This sequence is written at the end of the ciphertext
5. The End.

VIII. THE PROPOSED ALGORITHM FOR DECODING

1. Getting Started
2. Receive the encrypted message
3. Approval of the last number to know the number of characters of the original text, which is equal to the number of letters of the key.
4. Depending on the first digits of the first number, the sequence of the key letter and the number of the first number is known.
5. Convert key numbers to binary encoding.
6. Obtaining the flux function is represented by the number that follows the key sequence number.
7. Obtaining the coded text numbers, which are all the remaining numbers.
8. Converting cipher text numbers to binary encoding
9. Decoding the RC4 algorithm
10. Obtain the original express text
11. The End.

IX. THE RESULTS

1. ***Enter the text to be encoded:*** Initially the text to be encrypted was entered

-

Help Me

Then it was converted to ASCII, then to the dual system, and it became

000101001001010011000100010000010100010001000100

After that, the number of bits made up of the text is calculated: No. of bits = 48

2. ***Key generation:*** The Random Function was used to generate a random number whose length is the length of the text to be encoded after converting to the binary system.

111010110110101100111011101111101011101110111011

3. ***Random key test:*** Two random randomness test keys were constructed, the first function: calculates the number of units and the key beeps and returns the value of F1, zero in the case of zero, equal to 1 or one to one or one to one.

The second function: we enter the value of n, which represents the number of blocks required and was n = 10 and search for n is within the key string and returns the values of zero in the absence of the presence of n in the presence of n in the presence of n. In example, F2 = 0, it also returns either zero in the absence of a n-1 gap or one in the n-1 gap. For example: Fitness = F = F1 + F2 + F3 Fitness = 0. As a result, Objective F is F3 = 0.
Since the value of fitness is not equal to 3, which is an amplification function, we will use the genetic algorithm.

4. Use the genetic algorithm:

- Initially, a primary community is generated from individuals. The creation of the primary generation is the starting point. In the solution of the issue, most researchers in this field have indicated that the process of constructing the primary generation is carried out randomly, and is programmatically done by using the (rand) that passes the standard. The one and the number of individuals differ from one issue to another, depending on the type of issue:

```
101001011110000100000111111101000000010011110100
111011010101110111111111110111100000010011110010
011010000000100111101001010010111100010011110100
011011010101110111111111110111100000010011110010
```

- Fitness Construction of the Fitness function:

F	no. of chromosomes	F1	F2	F3

0	1	0	0	0
1	2	0	1	0
2	3	0	1	1
1	4	0	1	0

- Building the probability function: This function was built to find the possibility of contributing each of the sections in the following way:

Pro = Fitness / total Fitness.

Pro1 = 2/4 = 1/2 Pro2 = 1/4 Pro3 = 1/4 Pro4 = 0.

- Selection : In this research, the roulette wheel method was chosen to choose individuals from the current generation to produce a new generation. This was done by building *Sel function* to be input to this function matrix representing the pro probability is then generating a matrix of random numbers, and then values compared to the value of each of the random values with the values of the matrix pro use of ready-made function rand, and then create a new matrix *newpro*. Below are the chromosomes that will participate in the marriage process.

- Crossover Mating : A function has been built to marry after the individuals were selected from the primary generation to have a role in the generation of the next generation, the process of marriage begins through each new two individuals, including This research is relying on simple crossover mating, where a random number was generated and approved as a displacement within the chromosome, at which the interfering (intermarriage) procedure is performed.

0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111

0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111

0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0010 1111 0100 0000 1110 1101 1111 1111 1101 0101 1101 1110

0111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0010 1111 0100 0000 1110 1101 0110 1101 0010 1100 1111 1111

- Mutation: After the marriage process, the role of the mutation in changing the results that result from the marriage process is taken. The mutation ratio is equal to 0.01, and the mutation is represented by forming a ***mut*** function.

Fitness	**No. of chromosomes**
2	3
2	3

1111 1111 1110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 **0**110 1111 0000 0000 0110 1101 0010 1100 1111 1111
0111 1111 1110 1111 **1**000 0000 1111 1111 1101 0101 1101 1110
0010 1111 0100 0000 1110 **0**101 0110 1101 0010 1100 1111 1111

- Evaluating the new generation: After the new generation has been generated, its members are evaluated in the same way as the primary generation.
- Substitution : In this research, a method was adopted that takes into account all members of the generation of both types. Good and bad: 60% of good people and 40% of bad people were taken. Assuming that a chromosome is obtained that matches the state N = 10 and satisfies the functions , F1=1, F2=1, F3 = 1, as follows: F=F1+F2+F3

```
0000 0000 0110 0110 1111 1111 1101 0101 0001 1001 1111 0000
```

5. *The coding* process***:***

The encoding used is Cipher Stream RC4 as follows:

- The text to be encrypted:

```
0100 0100 0100 0001 0100 1000 0100 1100 0100 1001 0100 0001
```

- The key used in the encryption process of the camouflage function:

```
0000 0000 0110 1111 1111 1101 0101 0001 1001 1111 0000
```

- The output of the coding process i.e. after the completion of the RC4 process:

```
0100 0100 0010 0111 1011 0111 1001 1001 0101 0000 1011 0001
```

6. *Flux* function ***Secure Hash Algorithm (SHA):***

After applying the law related to this function, we have the following camouflage function:

0101 0101

After converting it, it is equal to 133, after converting the encrypted text, it was as follows:

68 39 183 153 80 177

Also, the key after it was converted was in the form:

00 102 256 213 25 240

In order for the text to be properly encoded and decoded, the coded text has to be available . The key to the recipient of the encrypted message also has the ability to open the encryption, and the key is hidden encoded inside the encrypted message. As a result of the coding, concealment and camouflage function, the text ready for transmission has become the following form, taking into consideration the change in the spaces between the numbers to zero to increase the camouflage:

```
00010206802560216058497043558786500657043067800034565600540230450
```

7. *Decryption:*

After receiving the encrypted text and according to the algorithm of decoding the last number represents the number of characters of the text encoded as well as the key which is 6. Also, the key locations are: 10, 8, 4, 3, 1, 0

It is: 00 102 356 213 25 240

The camouflage function is 133 to recalculate to verify the reliability of the file being sent, and the original text is: 68 39 183 153 80 177

After the conversion of the binary system and the XOR process, the result:

```
0100 0100 11011 1111 0011 0010 1101 0001 1110 0011 0101 1100
```

Then it produces: 68 65 72 76 73 75

After converting it to the characters, the result was: help me, which is the original text.

8. Ciphering and decoding time:

The speed of implementation of the coding and coding algorithm was measured to ensure its speed and results. Shown in the following table, given that the program has been applied to a computer with high specifications,

Labtop acer

- Intel Celeron M processor 430 (1.73 GHz, 533 MHz FSB, 1 MB L2 cacha)
- Intel Graphics Media Accelerator 950

The length of the encoding text	**Coding Time**	**Decoding Time**
20 character	0.26574	0.12344
40 character	0.254867	0.18675
80 character	0.35044	0.29288
100 character	0.457685	0.38576

We notice from the table above that the coding time for texts and their decoding are good and they are not subject to any rule because the coding of the scripts relies on the key generation process using the genetic algorithm. We also note that the coding time is less than the coding time because in decoding the code is not used genetic algorithm.

X. CONCLUSIONS AND RECOMMENDATIONS

The proposed method for improving encryption using streamlined encoding has the potential to:

Implementation on any computer that has the Matlab system in place, in addition to the difficulty in obtaining cipher key within ciphertext. Also, the proposed method is confidential because of the randomness of the key, which leads to hiding the statistical properties of the express text language and knowing part of the key sequence is not useful in knowing all of the sequences are not repeated as in the known linear and non-linear displacement registers. Therefore, the method has the advantage of being proven in front of a well-known attack (plantext). Other smart technologies can also be used to generate the key, such as the use of neural networks. In addition to the possibility of merging more than one encryption algorithm and utilizing the genetic algorithm by generating key, and use the known flux function which fulfills the specifications required for the flux function MD5.

REFERENCES

1. R. V Ericson, Crime in an insecure world. Polity, 2007.
2. D. H. Flaherty, “Protecting privacy in police information systems: data protection in the Canadian Police Information Centre,” Univ. Tor. Law J., vol. 36, no. 2, pp. 116–148, 1986.
3. M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, “A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements,” Requir. Eng., vol. 16, no. 1, pp. 3–32, 2011.
4. E. McFadzean, J. Ezingeard, and D. Birchall, “Perception of risk and the strategic impact of existing IT on information security strategy at board level,” Online Inf. Rev., 2007.
5. S. Glass, T. Hiller, S. Jacobs, and C. Perkins, “Mobile IP authentication, authorization, and accounting requirements,” Work Prog., 2000.
6. R. M. Sharma and P. Choudhary, “Synthesis and Simulation of FPGA Based RC4 Encryption Method,” Synthesis (Stuttg)., vol. 5, no. 4, 2016.
7. A. A. Alhijaj and M. Kamil Hussein, “Stereo Images Encryption by OSA & RSA Algorithms,” J. Phys. Conf. Ser., vol. 1279, no. 1, 2019.
8. G. S. Basheer, “Application of Polyalphabetic Substitution Cipher using Genetic Algorithm,” AL-Rafidain J. Comput. Sci. Math., vol. 5, no. 1, pp. 57–68, 2008.
9. A. Gorodilov and V. Morozenko, “Genetic Algorithm for finding the Key’s length and Cryptanalysis of the Permutation Cipher,” 2008.
10. R. Toemeh and S. Arumugam, “Breaking transposition cipher with genetic algorithm,” Elektron. ir Elektrotechnika, vol. 79, no. 7, pp. 75–78, 2007.
11. A. Dimovski and D. Gligoroski, “Attacks on the transposition ciphers using optimization heuristics,” Proc. ICEST, pp. 1–4, 2003.
12. M. K. Hussein and A. A. Alhijaj, “TDL and ron rivest, adi shamir and leonard adleman in stereo images encrypt,” J. Adv. Res. Dyn. Control Syst., vol. 11, no. 1 Special Issue, pp. 1811–1817, 2019.
13. F. J. Kherad, H. R. Naji, M. V Malakooti, and P. Haghighat, “A new symmetric cryptography algorithm to secure e-commerce transactions,” in 2010 International Conference on Financial Theory and Engineering, 2010, pp. 234–237.
14. A. Maximov, Some words on cryptanalysis of stream ciphers. Citeseer, 2006.
15. D. Eastlake and T. Hansen, “US secure hash algorithms (SHA and HMAC-SHA).” RFC 4634, July, 2006.
16. R. Patel and N. Chaudhary, “Analyzing Digital Signature Robustness with Message Digest Algorithms,” Comput. Appl. Commun. Secur., 2012.
17. I. Mironov, “Hash functions: Theory, attacks, and applications,” Microsoft Res. Silicon Val. Campus.

Noviembre, 2005.

18. N. Ferguson et al., “The Skein hash function family,” Submiss. to NIST (round 3), vol. 7, no. 7.5, p. 3, 2010.
19. R. S. McIntyre et al., “Cognitive deficits and functional outcomes in major depressive disorder: determinants, substrates, and treatment interventions,” Depress. Anxiety, vol. 30, no. 6, pp. 515–527, 2013.
20. C. Neudecker, N. Mewes, A. K. Reimers, and A. Woll, “Exercise interventions in children and adolescents with ADHD: a systematic review,” J. Atten. Disord., vol. 23, no. 4, pp. 307–324, 2019.