ICIC Express Letters
Volume **12**, Number **2**, February **2018**
 ISSN 1881-803X
pp. **117–123**

A NEW ADVANCED CRYPTOGRAPHIC ALGORITHM SYSTEM FOR BINARY CODES BY MEANS OF MATHEMATICAL EQUATION

Mohamed Sherif Ksasy[1], Ali Elsherbeni Takieldeen[2]
Samaa Mohamed Shohieb[3] and Ahmed Hussein Eltengy[4]

[1]Faculty of Engineering
[3]Faculty of Computers and Information Sciences
Mansoura University
60 Elgomhoria St., Mansoura 35516, Egypt
msmksasy@gmail.com; sm.shohieb@yahoo.com

[2]Faculty of Engineering
Delta University
Costal International Road, Industrial Area, Gamasa 11152, Egypt
a_takieldeen@yahoo.com

[4]Department of Communication
Alexandria University
Lotfy El-Sied st. off Gamal Abd El-Naser, Alexandria 11432, Egypt
tengy_fox@yahoo.com

Received September 2017; accepted November 2017

Abstract. *The transmitted and received data through the communication channels and the stored data on media must be secured. There are many cryptographic algorithm systems such as Advanced Encryption Standard (AES), Blowfish and Rivest, Shamir, and Adleman (RSA). Trying to strengthen these algorithms faces many drawbacks, and the most important problem is adding much time to the delay packet to sustain security on the communication channel between terminals. This paper proposes an algorithm system for binary encryption and decryption. This algorithm based on converting zero to one and one to zero based on a mathematical equation. The proposed algorithm introduces high performance of speed in addition to reproducing the original binary data bits with no loss of information in the decryption and encryption process. Also this algorithm is a very fast, precise, distinguished method and more flexible than other recent algorithms and it has passed all security requirements to be used in many software and hardware applications; for example, the speed for encryption/decryption process after test is about 20% faster than the AES algorithm encryption/decryption process.*
Keywords: Binary codes, CryptoBin, Cryptography, Decryption, Encryption, Security analysis

1. **Introduction.** Cryptography has a long history in securing data and information and keeping it away from lost or from being available for others and intruders during transmitting it across insecure networks such as (Internet or wireless networks) or storing it on media. So that it was mandatory to hide or change the plaintext form that cannot be read by anyone except who has the authorization to read it [1]. The cryptography system is a set of processes or functions with keys to convert the plaintext to encrypted text and then convert it back to the original text again [2].

The traditional cryptographic algorithm systems have many drawbacks [3], such as Advanced Encryption Standard (AES) [4] which needs more processing and requires more rounds of communication, Blowfish [5] which also has weakness in decryption process over other algorithms in terms of time consumption and serially in throughput, the Rivest, Shamir, and Adleman (RSA) [6] encryption and decryption algorithm which needs a lot of calculation and whose speed is slow, compared with the symmetric algorithm [7].

In this paper a new cryptographic algorithm system for binary codes based on a proposed mathematical equation. The main idea of this algorithm is converting zero to one and one to zero by using a mathematical equation using the values, which generated from the inserted two keys by mathematical function using based on logic functions. The message is divided into bytes, each of which is composed of 8 bits. The first key specifies the bit where the change of bit value (0 to 1, 1 to 0) occurs till the end of each byte of the message. The second key value is used to calculate how many bytes used with the first key value before it decreased by one-bit value. The decryption procedure is similar in process to the encryption in the same order. The proposed algorithm system can regenerate the original binary data byte with no loss of data or information for the encryption and decryption process. By using two secret keys, the algorithm is more secure and it is hard to guess the keys value or attacked. The proposed algorithm has been compared with other recent algorithms and it was fast, simple and flexible enough. For validation of our new algorithm the security requirements have been applied and it has been suitable for using it in many software and hardware applications.

The second section demonstrates the problem formulation, the proposed algorithm (CryptoBin) will be explained in the third section, the fourth section discusses the statistical tests for cryptographic applications and its results, and finally the conclusion will be introduced.

2. **Problem Formulation.** Cryptographic algorithms are used for keeping records secured. Their complexity and capacity to resist attack varies from one system to another such as Triple Data Encryption Algorithm (3DES) [8], Rivest Cipher 4 (RC4) [9], Advanced Encryption Standard (AES) [5], Blowfish [6], and Rivest, Shamir, and Adleman (RSA) [6].

Information is one of the most important elements of any organization, we do not find enough attention to protect it, and the lack of attention may be due to a lack of knowledge of the technology that keeps this information (Computer), or just think that this information is impossible to be lost [10]. With the development of computer systems and networks discovered and used extensively, it has become much more complicated to protect information [11].

Information can be divided into two main types: first one, Electronic Information and the second one, traditional information (such as paper) [12]. Electronic information is susceptible to damage and attack more than traditional information for the following reasons [13]:

- The possibility of electronic information leakage.
- Electronic information not visible to the eye.
- Information may save by using a small-sized liquid.
- The difficulty of dealing with the computer.
- Increased communication and networking.

Information security is used to protect and secure the information and all the resources used to process information, the security of computers, the mass transfer of information, and networks. In order to prevent security breaches of confidential information and save information in a secure way, that will be done by using a secure encryption system [14].

3. **Proposed CryptoBin Algorithm.** Due to the large number of global encryption systems and frequent attempts by hackers to break these systems, it has become necessary to invent a new encryption system characterized by speed, flexibility and ease in handling, as well as to reduce the risk of others to attack or obtain important information.

3.1. **Secret key generation.** The idea of new cryptographic system is dealing with a single 8-bits which represented as a byte where all digital signals are in form of $(0, 1)$. The system collects every 8 digital signals (8-bits) consisting of $(0, 1)$ until it has made up one byte. The algorithm uses two secret keys. The first secret key value can be derived from a simple mathematical equation to obtain a number from 0 to 7 which it used to allocate which bit of each byte the system will start swapping (0 to 1, 1 to 0) from. The first secret key value divides the 8-bits into two parts; the first part is before the value of the key, it has no change on it, and the NOT function is applied to the second part which comes after the value of the first key to the end of bits in the byte, so the zero becomes one and the one becomes zero. The first key value will be applied to a number of bytes equal to the second key value. The second key is used to divide the whole message into intervals, each of whose length equals the second key value. After each interval the first key value decreased by one number. Using two key values increases the security and makes it more difficult to guess the value of keys. The result from that encryption algorithm is having encrypted byte with different values from the original byte before encryption, without using complex equations which consume time. At decryption process the first and second secret keys are derived from the same simple mathematical equations and do the same job as encryption process. So the final result is obtaining the original byte again.

3.2. **CryptoBin structure.** Assume that the first byte in the message is 10111001 in binary value which is equivalent to 185 in decimal value, and we called it "Plain_binary", then a value for the first key will be inserted by the user, this value will be applied to the secret key equation (any number MOD (8)) to generate a new number from 0 to 7; for example, the generated number will be = 3, which will be used to allocate which bit of each byte the system will start swapping (0 to 1, 1 to 0) from. The user will insert another number that will be used to determine the interval length which we called the second key value. For example, the second key value = 100.

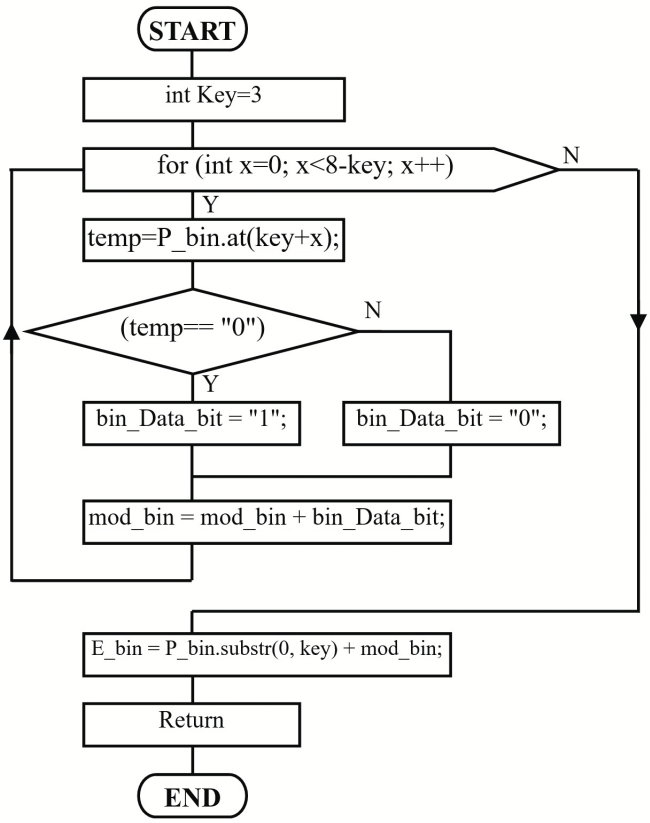

FIGURE 1. Flowchart for CryptoBin structure

Figure 1 shows a flowchart that describes the CryptoBin system from starting process, then receiving the keys values, after that starting a loop from 1 to 8, the received key value which archives the NOT function on bits, and creates a temporary value that stores the bit value, if it is 0 then swap it to 1 and vice versa. The next step is merging swapped values with non-swapped values to combine the encrypted byte.

```
'CryptoBin Software Code 1
Dim mod_bin, bin_Data_bit As String, key As Integer = 3, Plain_binary As string =
"10111001", Encrypted_binary As string
For x = 1 To 8 − key 'NOT Function
        If Mid(Plain_binary, key + x, 1) = 0 Then : bin_Data_bit = 1
        Else       bin_Data_bit = 0 : End If
        mod_bin = mod_bin + bin_Data_bit
Next x
Encrypted_binary = Mid(Plain_binary, 1, key) + mod_bin : mod_bin = vb.empty 'reset
value
```

the Result is ***Encrypted_binary = 10100110 = 166*** in decimal value

The first example can be used in stream binary encryption and the same software code can be used also in block of binary values with the next code example

```
'CryptoBin Software Code 2
Dim Plain_binary = {"10001110", "10011011", "00010010"}, mod_bin, bin_Data_bit,
temp_binary As String, key As Integer = 3, Encrypted_binary As New ArrayList
For n = 0 To Plain_binary.Count − 1 'Encryption
For x = 1 To 8 − key
        If Mid(Plain_binary(n), key + x, 1) = 0 Then : bin_Data_bit = 1
        Else       bin_Data_bit = 0 : End If
             mod_bin = mod_bin + bin_Data_bit
Next x
temp_binary = Mid(Plain_binary(n), 1, key) + mod_bin : Encrypted_binary.Add(temp_
binary) : mod_bin = vb.empty : Next n
Decrypted_binary As New ArrayList 'Decryption
For n = 0 To Encrypted_binary.Count − 1
For x = 1 To 8 − key                    'NOT Function
        If Mid(Encrypted_binary(n), key + x, 1) = 0 Then : bin_Data_bit = 1
        Else bin_Data_bit = 0 : End if
        mod_bin = mod_bin + bin_Data_bit : Next x
        temp_binary = Mid(Encrypted_binary(n), 1, key) + mod_bin
        Decrypted_binary.Add(temp_binary) : mod_bin = vb.empty
Next n
```

Figure 2(a) shows that the secret key divides the byte into two parts; the first part has no change in bits' value, but the second part the NOT function is applied to it, that means the bit's value changes (0 to 1, 1 to 0) till the end of each byte of the message. This results in a new byte value differing from the original byte value called encrypted byte.

Figure 2(b) shows that the secret key divides the encrypted byte into two parts; the first part has no change in bits' value, but the second part the NOT function is applied to it, that means the bit's value changes (0 to 1, 1 to 0) till the end of each byte of the message. This results in original byte value differing from the encrypted byte value called decrypted byte.

4. **Statistical Tests for Cryptographic Applications and Results.** The statistical tests consist of sixteen types of tests that were created to check the randomness of binary sequences. These tests concentrate on an assortment of various sorts of non-randomness

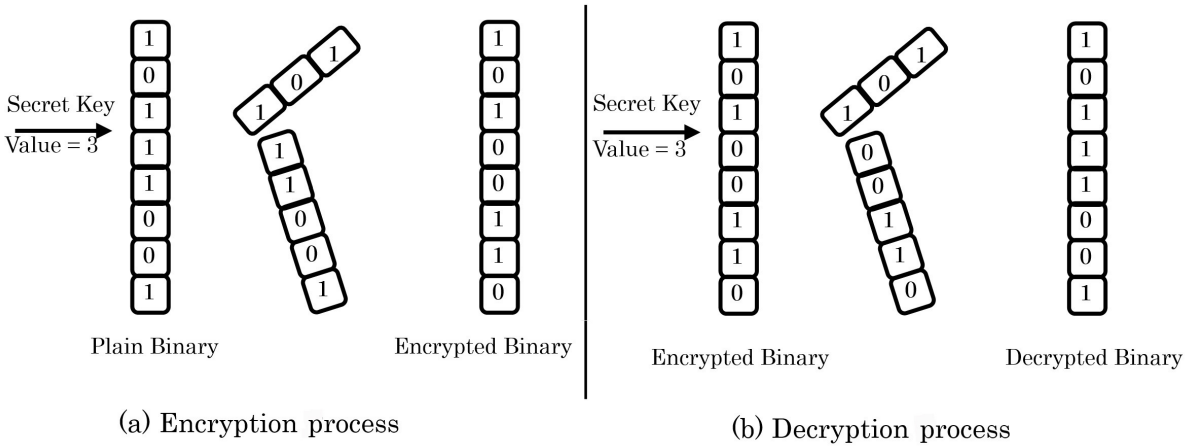

FIGURE 2. Encryption and decryption processes

that might exist in sequences. Each test is depending on a calculated test statistic value, which is an element of the testing sequence. The test measurement value is used to calculate a P_{value} that outlines the quality of the proof against the theory about flawless arbitrariness of the number generator. For each test from test suite, the P_{value} is the likelihood that the ideal arbitrary number generator would have created a sequence less random than the sequence that was tested, given the sort of non-arbitrariness surveyed by the test. In the event that a P_{value} for a test is resolved to be equivalent to one, then the grouping seems to have idealized irregularity. A P_{value} of zero shows that the grouping seems, by all accounts, to be totally non-arbitrary. A critical level can be decided for the tests. The parameter indicates the likelihood of test dismissing a testing generator that is in certainty superbly arbitrary. If $P_{value} \geq \alpha$ then the sequence appears to be random to significant level α. If $P_{value} < \alpha$ the sequence appears to be non-random. The typical values for α are chosen in the range [0.001-0.01] [15].

In this algorithm technique Frequency (Monobit) Test is used and the P_{value} for different file sizes as shown in Table 1 and Figure 3 shows the same results of this test.

TABLE 1. Frequency test results

File Size	1KB	10KB	100KB	1MB	10MB
P_{value}	0.8975	0.961	0.973	0.9757	0.9884

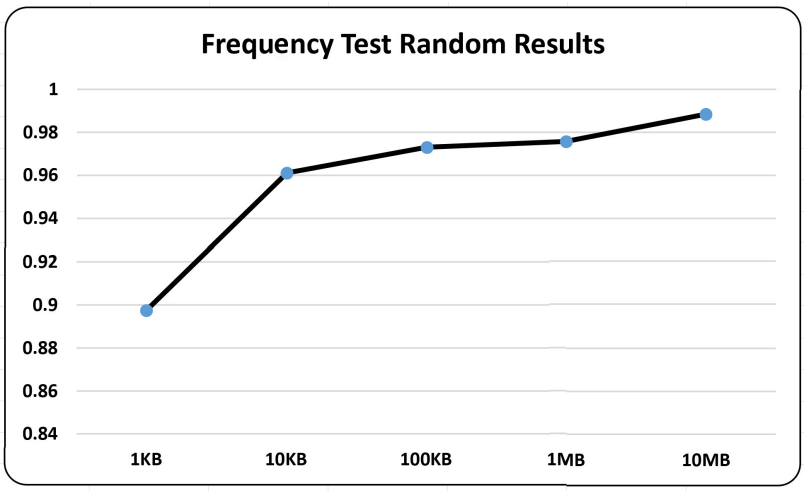

FIGURE 3. Frequency test random results

Test file size is (1kB = 1024Byte = 8192Bit)

$$n_0 = 4062,\ n_1 = 4130,\ n = 8192,\ S_n = 68$$
$$S_{obs} = \left(|S_n|/\sqrt{n}\right) = (68/90.509668) = 0.7513$$
$$P_{value} = erfc\left(S_{obs}/\sqrt{S_n}\right) = 0.8873$$

$P_{value} \geq 0.01$ then the sequence is **random**.

The results came out from frequency test prove that the generated encrypted file from using CryptoBin Algorithm has a random sequence.

By implementing algorithms in a standard language such as (Java, C++), using their standard specifications, and testing on two different hardware platforms, compare their speed performance in encryption/decryption process by testing data files in different sizes (32, 64, 128, 256, 512) KB and measure the time duration of each process separately. Table 2 shows the encryption process time/file size for each algorithm. As a result, the new algorithm is at least 20% faster than all of them.

Figure 4 is an illustrative example of a sample of 128, 256 and 512 KB clusters measuring the speed of the cryptographic process for each algorithm, demonstrating the speed of the new algorithm.

From previous comparison results, it has been revealed that the algorithm is relatively fast and simple to use. It fits hardware and software applications which depend on data streaming.

The algorithm does not affect the continuous streaming of data when interrupting it; furthermore, it maintains high speed streaming data in highly secured environment.

Table 2. Comparison of popular algorithms

Data (KB)	Encryption/Decryption Time (seconds)				
	RSA	RC4	Blowfish	AES	CryptoBin (ours)
32	9.45	4.60	4.52	2.02	0.14
64	10.53	4.88	4.70	2.13	0.27
128	11.41	5.20	5.02	2.29	0.448
256	16.27	6.70	6.51	2.47	1.70
512	24.44	7.40	7.24	2.63	2.20

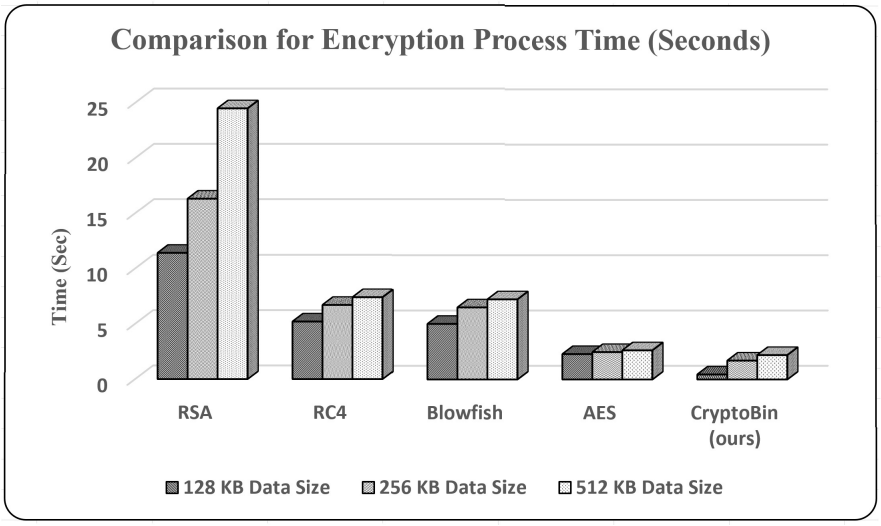

Figure 4. Comparison chart of popular algorithms and CryptoBin

5. **Conclusions.** In this paper, we have presented a new algorithm for binary encryption based on $(0, 1)$ called CryptoBin. A simple technique has been used in this algorithm; it depends on dealing with each binary byte $(0, 1)$ separately, the input data size is one byte and the output data size is also one byte so that the encrypted message is equal to the plain message. The proposed algorithm has two secret keys, that increases the security in the encryption process and also it is applied by the same steps to the decryption process and it takes the same time duration and also introduces high security. This system can be used in many applications such as encrypting a block of data and storing it as a file, and also it can be used in hardware systems and embedded systems, such as microcontrollers which deal with a stream of digital data to encrypt data byte by byte. The advantages of this system can be concluded in simplicity, speed, flexibility, and ease-of-use. And using two keys values increases the complexity and security for this algorithm. Non-proliferation and known as new system makes it difficult to break. From the test results and comparing it with other algorithms, the CryptoBin algorithm is 20% much faster than the others in encryption/decryption process.

Acknowledgment. This work is partially supported by Prof. M. S. Ksasy, Dr. Ali E. Takieldeen and Dr. Samaa M. Shohieb, and we also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] K. Sharma and N. Bahl, Taxonomy of cryptography techniques for network security, *International Journal of Engineering and Computer Science*, vol.5, no.8, pp.17787-17793, 2016.

[2] E. A. Mohammed, N. F. Areed, A. Takieldeen and M. Abd-Elazeem, Novel cryptographic algorithm for 4G/LTE-A, *IJCA International Journal of Computer Applications*, vol.163, 2017.

[3] O. M. A. Al-Hazaimeh, A new approach for complex encrypting and decrypting data, *International Journal of Computer Networks & Communications (IJCNC)*, vol.5, no.2, 2013.

[4] D. Selent, Advanced encryption standard, *Rivier Academic Journal*, vol.6, no.2, 2010.

[5] S. Manku and K. Vasanth, Blowfish encryption algorithm for information security, *ARPN Journal of Engineering and Applied Sciences*, vol.10, no.10, 2015.

[6] A. E. T. El_Deen, E.-S. A. El-Badawy and S. N. Gobran, Digital image encryption based on RSA algorithm, *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, vol.9, no.1, 2014.

[7] P. Kumar and N. S. Rajaanadan, Data encryption and decryption using by triple DES performance efficiency analysis of cryptosystem, *International Journal of Innovative Research in Computer and Communication Engineering*, vol.4, no.3, 2016.

[8] N. Aleisa, A comparison of the 3DES and AES encryption standards, *International Journal of Security and Its Applications*, vol.9, no.7, pp.241-246, 2015.

[9] T. D. B. Weerasinghe, Analysis of a modified RC4 algorithm, *International Journal of Computer Applications*, vol.51, no.22, 2012.

[10] Y. Cherdantseva and J. Hilton, Information security and information assurance, in *Organizational, Legal, and Technological Dimensions of Information System Administration*, I. M. Portela and F. Almeida (eds.), IGI Global Publishing, 2013.

[11] H. M. El Bakry, A. E. Takieldeen and A. H. Eltengy, A new mobile application for encrypting SMS/multimedia messages on Android, *International Journal of Scientific & Engineering Research*, vol.4, no.12, 2013.

[12] H. M. El Bakry, A. E. Takieldeen and A. H. Eltengy, Implementation of an encryption scheme for voice calls, *International Journal of Computer Applications*, vol.144, no.2, 2016.

[13] A. Nigam and V. Singh, A study on data transmission security threats in cloud, *International Journal of Innovative Research in Computer and Communication Engineering*, vol.4, no.5, 2016.

[14] H. M. El Bakry, A. E. Takieldeen and A. H. Eltengy, Implementation of a hybrid encryption scheme for SMS/multimedia messages on Android, *International Journal of Computer Applications*, vol.85, no.2, 2014.

[15] A. Rukhin and J. Soto, A statistical test suite for random and pseudorandom number generators for cryptographic applications, *National Institute of Standards and Technology*, 2010.