A Novel Hybrid DNA Cryptographic System Using Symmetric Algorithm and Dynamic DNA Sequence Generator

Animesh Hazra and Ishani Roy

Abstract By the end of the twentieth century, an innovative technique for securing data was introduced called DNA Cryptography setting a new horizon in the security domain based on the concepts of DNA computing. In this paper, an enhanced version of the Blowfish algorithm combined with the Base64 encoding system and dynamic DNA sequence generator is proposed. Blowfish's versatile usability along with the incorporation of the Sequence generator's dynamicity introduces randomness in the whole security system making it intrusion-free. The proposed system's lower computation time and less complicated approach make it almost an ideal procedure for incorporating in security fields.

Keywords Base64 · Blowfish · DNA · Dynamic · OTP · Sequence generator

1 Introduction

In the field of cryptography, apart from the implementation of various symmetric algorithms, cryptographers realized the requirement of building asymmetric algorithms. Asymmetric algorithms are believed to be secure than most symmetric algorithms but in the case of speed, they are much slower. As conclusion, it can be said that cryptographers are constantly working on inventing new techniques. In the year 1994 Leonard Max Adleman one of the persons behind the invention of the RSA algorithm [1] invented the foundation concepts of DNA Cryptography. Since then, this technique is considered very promising in the domain of security for its numerous attractive features. So, in this paper, an innovative and simple cryptographic system has been proposed. The suggested system is unique because no matter how much security other systems can provide, only a very few numbers of them can be able to

A. Hazra
Jalpaiguri Government Engineering College, Jalpaiguri, West Bengal, India
e-mail: animesh.hazra@cse.jgec.ac.in

I. Roy (✉)
Cooch Behar Government Engineering College, Cooch Behar, West Bengal, India

B. Sikdar et al. (eds.), *Proceedings of the 3rd International Conference on Communication, Devices and Computing*, Lecture Notes in Electrical Engineering 851,
https://doi.org/10.1007/978-981-16-9154-6_39

do that in a very less time-consuming manner. It is needless to say in the case of a double layer security system, robustness will be provided very well. Blowfish is an algorithm which is known for its strong security incorporation power and with this, our special dynamic sequence generator makes the message decryption work for any intruder impossible. Last but not least this security system is able to deliver another added advantage that can be seen in very few cryptographic systems, convenient to use. Not all security systems are able to come up with such praiseworthy qualities all encapsulated together in a single system. At the end of the paper, in the security analysis of the proposed algorithm section, a brief analysis of its performance is presented. Before proceeding to the discussion of the proposed method, in the technical background section, a thorough review of the used techniques is done in a systematic manner.

2 Technical Background

The encryption mechanisms are mainly divided into two categories that are symmetric algorithm and asymmetric algorithm [2]. Symmetric algorithms use an identical key for encryption and decryption purposes whereas asymmetric algorithms use different keys for encryption and decryption purposes (public key for the encryption at sender end and private key for the decryption at the receiver end). All the basic concepts on which the proposed algorithm depends on are discussed below in detail as follows.

2.1 Blowfish Algorithm

It is one of the famous symmetric algorithms [3] invented by Bruce Schneier and block cipher as well as a Feistel cipher of 16 rounds. Unlike other algorithms, it has a variable key size ranging from 32 to 448 bits. It works on a 64-bit block size. The algorithm works in two stages. In the first stage, from the key bits, several sub-keys are generated. P-arrays consist of 18 entries each, having 32-bit sub-keys and four 32-bit S-boxes having 256 entries each, are initialized with a fixed string of hexadecimal digits of pi (mathematical constant π). After initialization, p1 is bitwise XOR-ed with the first 32 key bits, p2 is bitwise XOR-ed with the second 32 key bits and so on until all the key bits are XOR-ed successfully (possible up to p14). For p15 to p18, first to fourth key bits are reused for bitwise XOR operation. Next, the sub-keys (p-arrays and S-boxes are referred to as sub-keys) are used on a 64-bit block initialized with every bit value 0 to run the Blowfish encryption process. The resultant 64 bit is halved and each 32-bit replaces the prior 32-bit values of p1 and p2 respectively. The resultant values are again encrypted with modified sub-keys. Now, the resulting 64 bit is halved and replaces the prior values of p3 and p4 respectively. In this fashion up to p18 and after p-arrays, all previous values of S-boxes are changed.

2.2 Base64 Encoding Scheme

Base64 is prevalent among many binaries to text encoding schemes in the cyber world. Here, binary data is converted into ASCII string format. A specific MIME (Multipurpose Internet Mail Extension) content transfer encoding is the origin of the term "Base64". It translates the binary data into a radix 64 representation. Each Base64 digit represents the 6-bit binary data. It is used to embed binary data like image files into text formats like HTML and CSS files.

2.3 One-Time Pad (OTP)

One-time pad, also known as Vernam cipher or the perfect cipher is one of the renowned cryptographic methods where a random secret key (referred to as one-time pad) of at least the same size or longer in length than the message to be sent is generated. Once generated key cannot be reused. The plaintext is paired by having bitwise or character-wise modular operation to the corresponding each bit or character of the generated key. There are only two copies of the generated key, one for the sender and another for the receiver. It can be presented with the help of a simple equation as follows:

$$C_p = B_p \odot O_p (p = 0, 1, 2, 4 \cdots n). \tag{1}$$

Here, C_p is the resultant ciphertext, B_p is the bitwise binary value, O_p is the bitwise one-time pad (OTP) value, and $\odot$ is the XNOR operation. Infinite time as well as infinite computations cannot break this encryption mechanism, as it is mathematically impossible.

2.4 DNA Computing

It is a branch of computing that uses DNA molecules, biochemistry, and other biological background-based hardware. Leonard Adleman in 1994 invented the DNA computing concept [4]. From that time this approach has been implemented in various applications, i.e., developing GPS systems, recognition systems combined with artificial intelligence (AI), etc. Therefore, it can be concluded that DNA is well suited for data processing. Some of its numerous features are listed below as follows:

(a) 1 g DNA can hold 10^8 TB of data.
(b) As specified by Adleman DNA strand combined computations made the calculations speed far better than the fastest computers (nearly 100 times faster).
(c) There is no requirement of external power sources as chemical bonds are the basis of DNA formation.

DNA molecules process different types of combinations at once and as a result, developing various characteristics from it is the basic point of its parallel form of computing.

3 Literature Survey

It is recommended to discuss about some previously suggested security systems by other authors to develop a concept regarding the presented work approach as well as the pros and cons of those suggested methodologies which is also shown in Table 4.

Raj et al. [5] developed an innovative DNA cryptographic procedure based on a symmetric algorithm where the input plain text is converted into ASCII values followed by binary values and then DNA base sequences. A modular cyclic operation is done using that private key on them producing the final encrypted text. Shanmugasundaram et al. [6] introduced a new DNA encryption procedure based on cellular automata where input plain text is converted into DNA base sequences followed by binary bitstream generation. Hassan Al-Mahdi et al. [7] developed a unique DNA security system. The input plain text is converted into ASCII values followed by binary bits. Finally, binary groups are converted to hexadecimal values to represent the final ciphertext. In the paper of Kamaraj et al. [8], the authors developed a DNA cryptographic algorithm based on the FPGA and Vigenere cipher concept. The input plain text is given through FPGA to convert the text into triplet codons as per the table recommended by the authors. Finally, the codons are encrypted by XOR operation with generated key followed by Vigenere cipher.

4 Proposed Methodology

An ideal security system consists of three major constituents, i.e., speed, security, and key generation. To ensure security it is recommended to work on versatile key generations. To increase feasibility, the whole security system should respond quickly. To overcome the shortcomings mentioned earlier in the literature survey section, in this paper Blowfish algorithm has been implemented to introduce a more convenient way than the past security schemes along with a unique dynamic sequence generator to create randomness in the entire system. This enhanced, highly secured, and ideal procedure is discussed in depth further below.

4.1 Proposed Encryption Procedure

Step 1. A plain text (P) is taken as input.

Step 2. The text is encoded using the Blowfish algorithm in CFB (Cipher Feed-Back) mode. In this mode, each ciphertext block gets fed back into the encryption process in order to encrypt the next plain text block.
Step 3. The encrypted text (C_k) is now encoded using the Base64 encoding scheme which is discussed earlier in sub-Sect. 2.2.
Step 4. Now, the encoded string obtained previously is converted to ASCII values.
Step 5. Each digit of the ASCII value string obtained is now transformed into 4-bit binary values.
Step 6. Now, a complement operation on each bit of binary bitstream is made.
Step 7. Complemented binary bitstream obtained in the previous step is now converted to hexadecimal values.
Step 8. Each hexadecimal value produces different DNA base sequences from the list at a different time as recommended in the sequence generator table illustrated in Table 1. Suppose, the hexadecimal digit is 6, then the sequencer can generate CATA or CATG or any other base sequences for the value 6 shown in Table 1.
Step 9. At present, the DNA base sequences are converted to binary bitstreams (B_p) as per the scheme described in Table 2 as follows.
Step 10. Now, the previously obtained binary bit stream is XNOR-ed with another random generated binary bit stream of same length (O_p) or OTP.
Step 11. The resultant binary bit stream (C_p) is finally converted into the DNA base sequences (cipher-text) as per the scheme described in Table 2.

4.2 Flowchart of Proposed Algorithm

To make the whole decryption procedure handy, it can be easily done only in reverse order. In Fig. 1 below, a flowchart of encryption as well as decryption procedure is presented in detail.

5 Discussion

The setup requirements as well as the security analysis of the algorithm proposed are illustrated below in detail.

5.1 Specifications of Computing Processor and Software Used

The proposed security system is built using the Python 3.7 programming language on Spyder 3.3.2 IDE. The device used for developing this security scheme consists of an Intel Core i5 7th Generation processor with 2.71 GHz speed, 1 TB of Hard

Table 1 Illustration of the hexadecimal digit to random DNA base sequence generation

Hexadecimal digit	Possible 16 DNA sequences corresponding to each hexadecimal digit							
0	GCTA GCCA	GCAA GCGA	GCTT GCCT	GCAT GCGT	GCTC GCCC	GCAC GCGC	GCTG GCCG	GCAG GCGG
1	CGTA CGCA	CGAA CGGA	CGTT CGCT	CGAT CGGT	CGTC CGCC	CGAC CGGC	CGTG CGCG	CGAG CGGG
2	AGAA AGGA	AATA AACA	AGAT AGGT	AATT AACT	AGAC AGGC	AATC AACC	AGAG AGGG	AATG AACG
3	GATA GACA	TGTA TGCA	GATT GACT	TGTT TGCT	GATC GACC	TGTC TGCC	GATG GACG	TGTG TGCG
4	CAAA CAGA	GAAA GAGA	CAAT CAGT	GAAT GAGT	CAAC CAGC	GAAC GAGC	CAAG CAGG	GAAG GAGG
5	GGTA GGCA	GGAA GGGA	GGTT GGCT	GGAT GGGT	GGTC GGCC	GGAC GGGC	GGTG GGCG	GGAG GGGG
6	CATA CACA	ATTA ATCA	CATT CACT	ATTT ATCT	CATC CACC	ATTC ATCC	CATG CACG	ATTG ATCG
7	ATAA ATGA	TTAA TTGA	ATAT ATGT	TTAT TTGT	ATAC ATGC	TTAC TTGC	ATAG ATGG	TTAG TTGG
8	CTTA CTCA	CTAA CTGA	CTTT CTCT	CTAT CTGT	CTTC CTCC	CTAC CTGC	CTTG CTCG	CTAG CTGG
9	AAAA AAGA	ATGA TTTA	AAAT AAGT	ATGT TTTT	AAAC AAGC	ATGC TTTC	AAAG AAGG	ATGG TTTG
a	TTCA CCTA	CCCA CCAA	TTCT CCTT	CCCT CCAT	TTCC CCTC	CCCC CCAC	TTCG CCTG	CCCG CCAG
b	CCGA TCTA	TCCA TCAA	CCGT TCTT	TCCT TCAT	CCGC TCTC	TCCC TCAC	CCGG TCTG	TCCG TCAG
c	TCGA AGTA	AGCA ACTA	TCGT AGTT	AGCT ACTT	TCGC AGTC	AGCC ACTC	TCGG AGTG	AGCG ACTG
d	ACCA ACAA	ACGA TGGA	ACCT ACAT	ACGT TGGT	ACCC ACAC	ACGC TGGC	ACCG ACAG	ACGG TGGG
e	TATA TACA	GTTA GTCA	TATT TACT	GTTT GTCT	TATC TACC	GTTC GTCC	TATG TACG	GTTG GTCG
f	GTAA GTGA	TAAA TGAA	GTAT GTGT	TAAT TGAT	GTAC GTGC	TAAC TGAC	GTAG GTGG	TAAG TGAG

Table 2 DNA base to binary bit conversion scheme

DNA base	Binary value
A	00
T	01
C	10
G	11

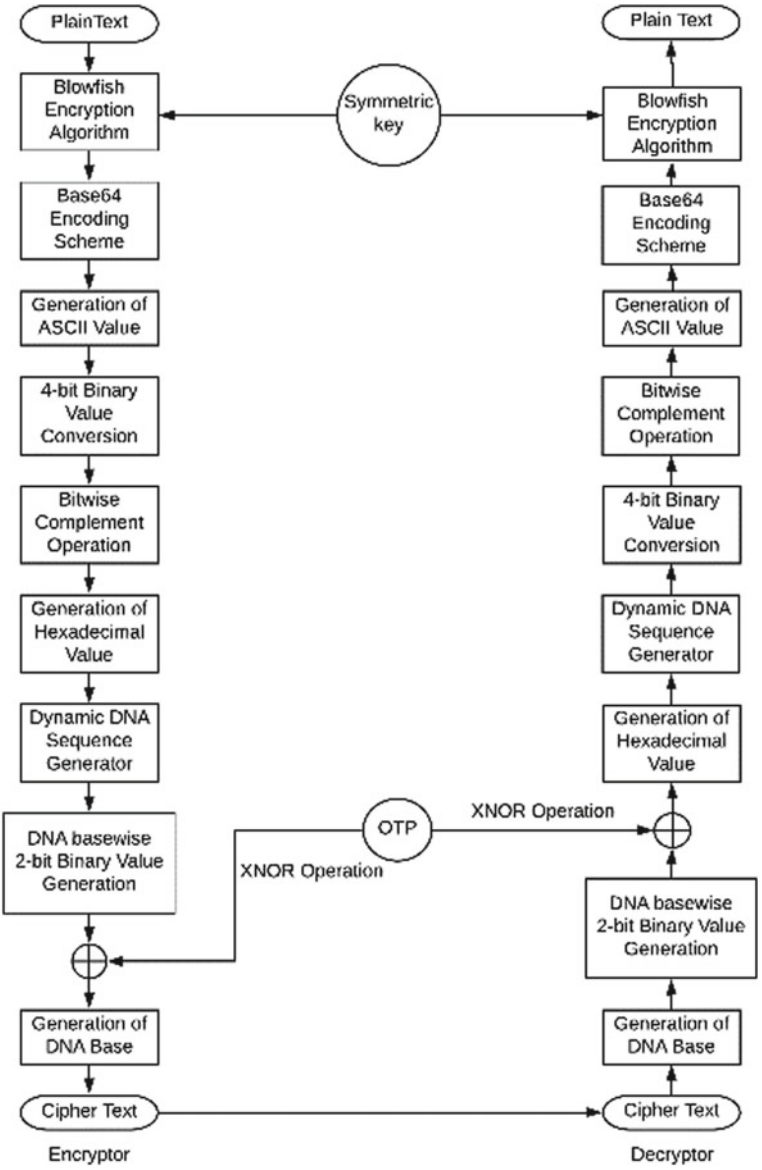

Fig. 1 Flowchart of the proposed encryption and decryption procedure

Disk Drive, and 8 GB RAM. The entire development was done on the Windows 10 (64-bit) Professional platform.

5.2 Security Analysis of the Proposed Algorithm

On examining the proposed methodology, complexities of the suggested algorithm as well as some salient security features are presented in this section as follows.

(a) Double layer security is incorporated in the presented algorithm. Plain text is encrypted using two keys, i.e., first time by symmetric key and the second time by a one-time pad.

(b) The time complexity of every step of the algorithm along with the proposed method is explained and shown in Table 3 where n is the length of the input string.

(c) Longer key length (up to 448 bits) in the Blowfish encryption procedure ensures stronger security but increases the computational complexity. It is the users choice so that they can generate how much longer the key length they want to use as per their requirements. Enabling users to generate different lengths of keys makes the entire algorithm highly flexible.

(d) One-time pad generation of n-bit long string has $O(2^n)$ computational complexity. Suppose there is an OTP, which is 4-bit long. To guess the correct OTP, at first 16 OTPs (0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111) needs to be generated, and then only they can be applied one by one to evaluate the correct one. With increasing the length of OTP, it becomes more difficult to assume the correct bit sequence without having any prior knowledge. Therefore, it is extremely difficult for cryptanalysts as well as supercomputers to successfully break the correct OTP.

(e) In this proposed algorithm, a dynamic DNA sequence generator incorporates another degree of security. For one hexadecimal value, there are 16 possible values available. Suppose four hexadecimal digits are taken, then the total possibility of DNA sequence generation is $16 \times 16 \times 16 \times 16$, i.e., 16^4. Therefore, if n digits are taken, then the total number of permutations will be 16^n. So, the overall permutation complexity will be $O(16^n)$, i.e., $O(2^{4n})$. It is an exponential time complexity algorithm and for a modest value of n it produces a huge value. Hence, we can say that the algorithm proposed here is highly secured and almost impossible to crack for the intruder.

5.3 Comparative Study

In this section, a comparative study is done based on the advantages and disadvantages of various encryption and decryption systems suggested by the researchers [9] Table 4.

Table 3 Time complexity evaluation of each step of the proposed algorithm

Name of the algorithm	Step no.	Steps of the proposed algorithm	Time Complexity of the associated step	The overall time complexity of the algorithm
Proposed encryption procedure	1	Blowfish encryption	O(n)	**O(n)**
	2	Base64 encoding	O(n)	
	3	Base64 to ASCII conversion	O(n)	
	4	ASCII to binary conversion	O(log n)	
	5	Complement operation	O(n)	
	6	The complement to hexadecimal conversion	O(n)	
	7	Dynamic DNA sequence generation	O(n)	
	8	OTP generation	O(n)	
	9	XNOR operation	O(n)	
	10	DNA base generation	O(n)	
Proposed decryption procedure	1	DNA base to 2-bit binary value generation	O(n)	**O(n)**
	2	XNOR operation	O(n)	
	3	Dynamic DNA sequence generation	O(n)	
	4	DNA sequence to hexadecimal conversion	O(n)	
	5	Hexadecimal to binary value conversion	O(n)	
	6	Complement operation	O(n)	
	7	Binary to ASCII value conversion	O(log n)	
	8	ASCII to Base64 encoding	O(n)	
	9	Base64 decoding	O(n)	
	10	Blowfish decryption	O(n)	

Table 4 Summary of some existing cryptographic techniques with the proposed methodology

Serial No.	Authors	Used methods	Advantages	Disadvantages
1	Babu et al. [10]	(a) Variable random length key (b) XOR operation	(a) Less storage space needed (b) Improved energy efficiency	(a) Distribution of key is really hectic
2	Kamaraj et al. [8]	(a) Vignere cipher (b) XOR operation	(a) Double layered security	(a) Security is too much dependent on the private key
3	Biswas et al. [11]	(a) Generation of the dynamic sequence table (b) RSA, ElGamal, and Paillier asymmetric encryption and decryption systems used	(a) Triple-layered security (b) Generation of the dynamic sequence table is mathematically impossible for the intruder	(a) Introducing different asymmetric cryptosystems increase the time complexity
4	Akkasaligar and Biradar [12]	(a) Pixel selection algorithm (b) Conversion to DNA encoded matrix as per the DNA base encoding rules	(a) Less time complexity (b) The security key is vast enough to resist the exhaustive attack	(a) Complex enough to implement in reality
5	Rahman et al. [13]	(a) Intron sequence generation (b) XNOR operation (c) Matrix manipulation	(a) Different cipher text is generated in each session (b) Can be adapted to the digital computing environment	(a) Requires to establish mutual authentication (b) Each time new encoding table generation is very hectic
6	Proposed Methodology	(a) Symmetric key (b) Dynamic DNA sequencer (c) XNOR operation	(a) Double layered security (b) User friendly for providing the scope of changing security key length according to user preference (c) Impossible to crack for its unique random nature	(a) Consumes more memory space in spite of small plain text

6 Conclusion and Future Scope

In this paper, one methodology is recommended where symmetric key exchange, OTP scheme, and dynamic DNA sequence generator give birth to a distinct hybrid cryptographic system. Cryptanalysts are unable to decipher the Blowfish encryption

procedure until date. Dynamic DNA sequencer makes the whole system so random that it is impossible for the intruder to break the cipher text without prior knowledge. Changing the key length in the Blowfish encryption algorithm as per the security requirements of the user as well as less complicated methods make this algorithm convenient to use in overall. By analyzing the whole encryption and decryption procedure, it can be concluded that it is theoretically as well as practically impossible to decrypt the cipher text hence ensuring the highest security possible in reality. In near future, implementing this algorithm in different applications and analyzing its complexity as well as feasibility is the utmost objective. Developing the proposed methodology accordingly on different mediums of data is another important agenda in the future. In case of smaller plain texts, the suggested algorithm generates much longer cipher texts that can be a setback in case of storage space utilization. Overcoming this hitch can also be considered as another future work for the existing methodology.

References

1. Pandey, M.K.: Implementation of DNA cryptosystem using hybrid approach. Res. J. Comput. Inf. Technol. Sci. **6**, 1–7 (2018)
2. Kahate, A.: Cryptography and Network Security, 3rd edn. New Delhi (2016)
3. Vadaviya, D.O., Tandel, P.H.: Secure encryption techniques using DNA computation. Int. J. Mod. Trends Eng. Res. **2**, 176–182 (2015)
4. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science **266**, 1021–1024 (1994)
5. Raj, B.B., Vijay, J.F., Mahalakshmi, T.: Secure data transfer through DNA cryptography using symmetric algorithm. Int. J. Comput. Appl. **133**, 19–23 (2016)
6. Shanmugasundaram, G., Thiyagarajan, P., Pavithra, S.: A novel DNA encryption system using cellular automata. Int. J. Secur., Priv. Trust Manage. **4**, 39–49 (2015)
7. Al-Mahdi, H., Shahin, O.R., Fouad, Y., Alkhaldi, K.: Design and analysis of DNA binary cryptography algorithm for plaintext. Int. J. Eng. Technol. **10**, 699–706 (2018)
8. Kamaraj, A., Bhrintha, A.P., Bhavithara, M.: DNA-based encryption and decryption using FPGA. Int. J. Curr. Res. Mod. Educ. 89–94 (2016)
9. Hazra, A., Ghosh, S., Jash, S.: A review on DNA based cryptographic techniques. Int. J. Netw. Secur. **20**, 1093–1104 (2018)
10. Babu, E.S., Raju, C.N., Prasad, M.HM.K.: Inspired pseudo biotic DNA based cryptographic mechanism against adaptive cryptographic attacks. Int. J. Netw. Secur. **18**, 291–303 (2016)
11. Biswas, M.R., Alam, K.M.R., Tamura, S., Morimoto, Y.: A technique for DNA cryptography based on dynamic mechanisms. J. Inf. Sec. Appl. **48**, 1–8 (2019)
12. Akkasaligar, P.T., Biradar, S.: Selective medical image encryption using DNA cryptography. Inf. Secur. J.: Glob. Perspect. **29**, 91–101 (2020)
13. Rahman, N.H.U., Balamurugan, C., Mariappan, R.: A novel DNA computing based encryption and decryption algorithm. Int. Conf. Inf. Commun. Technol. **46**, 463–475 (2015)