DNA-GENETIC ENCRYPTION TECHNIQUE

**Nitya Sree.P 17MIS1007**

Vellore Institute of Technology, Chennai

***Abstract***— In this paper , we propose the technique of DNA-Genetic Encryption (D-GET) to make the technique more reliable and less predictable. In this process, any form of digital data is binarized and transformed into DNA sequencing, reshaping, encrypting, crossover, mutating and then reshaping. D-GET 's main stages are repeated three times or more. Encrypted data is transmitted in a text or image file format. On the other hand, the receiver uses D-GET to decode and reshape the obtained data to the original data. This technique often converts the text into a picture and vice versa to improve security and numerous key sequences to increase the degree of diffusion and uncertainty, making it difficult to decode the resulting cypher data and making a perfect system of secrecy possible. Experimental findings show that the proposed technique has multilayer defence phases based on multi-stage and genetic operations against various attacks and a higher degree of security. Decrypted information is appropriate because of the total distinction between it and confidential information.

***Keywords***—DNATechnique,cryptography,bitexchange, Encrypt,Reshape,Mutation,Crossover,Genetic Algorithm.

1. **INTRODUCTION**

A modern cryptographical prototype is DNA cryptography. DNA is a nucleic acid which contains instructions for genetics. Adenine (A), cytosine ( C), guanine ( G) and thymine ( T) form the four bases present in DNA. Parallel processing capabilities are the greatest advantage of DNA cryptography. The approach to biomolecular cryptography based on DNA is planned. A new generation DNA-based key system is proposed to improve computation based on the DNA-based key expansion matrix using random key generation scheme speed. It proposes a novel and special technique based on biological simulation for DNA encryption and decryption. The plaintext is similarly split into two halves and translated for each session to DNA sequences using unique encoding table generation. After that, after applying suggested technique measures, the cypher text is produced. Some latest DNA Cryptography works are discussed and compared. Based on a secured symmetrical key generation , a new DNA cryptography algorithm is suggested. This encryption algorithm consists of encryption, random key generation and decryption in three steps. The text is translated to ASCII and then to DNA code at the encryption level. This initial cypher is translated to the final cypher by using random key-generated DNA sequences. The DNA-Genetic Encryption Technique (D-GET) is suggested in this paper. The hidden information was translated into binary data and then into DNA sequences in this technique. The D-GET is, moreover, an iterative algorithm. A round is called an iteration, and the number of iterations is three or more. There are four

operations in-round and it's iterative in nature. Iteration requires encryption, the method of reshaping and genetic operations. Furthermore, a symmetrical key is used. You may use any format of data type as secret data , i.e. text, word document, pixel image, audio , and video. Experimental studies indicate that reconstructed information is a standard copy of secret information. And they also show that the technique proposed maintains perfect security.

**II. IMPLEMENTATION OF D-GET TECHNIQUE**

In order to boost information security, this paper proposes the DNA-Genetic Encryption Technique (D-GET), which is an iterative algorithm. You may encrypt any form of data ( i.e., message,) Pre-processing, symmetric key encryption, reshaping and crossover and mutation are the principal stages of the proposed technique. They are clarified as follows.

1. *Pre-processing Stage*

This information has to be prepared after reading secret data, depending on its type. The ASCII values are translated in the case of a text file. Group them into 8 bit binary data. Each of the two adjacent bits is transferred to the four bases; adenine (A), cytosine ( C), guanine ( G) and thymine ( T), located in the DNA. According, for example, to Table 1.

In the event of text data are read into 8-bit binary data. Each of the two adjacent bits is transferred to the four bases; adenine (A), cytosine ( C), guanine ( G) and thymine ( T), located in the DNA.

Table 1. DNA and Representation of bits

|  |  |
| --- | --- |
| Bits | DNA |
| 00 | A |
|  |  |
| 01 | C |
| 10 | G |
|  |  |
| 11 | T |

A binary form (message, image , video or signal) may represent any type of data. The binary data is split into 8-bit classes. The four bases are moved to each of the two adjacent bits; A, C, G and T are found in DNA. Consider some hidden data as a binary bit file for better comprehension. Decompose the bit file into a group of bytes of any size. For example in the case:

Binary form of secret data:

………… 10 10 00 01 11 10 00 10 00 11 ………….

Convert Binary form to DNA bases:

* + GGACTGCGAT………….

*B. Encryption stage*

Symmetric and asymmetric key algorithms are two categories of cryptographic algorithms. A common key is exchanged between the sender and the recipient in the Symmetric Scheme. Asymmetric schemes have mathematically connected keys that are public and private. High-speed cryptography technology is the key advantage of the symmetric cryptographic algorithm and is more suitable for encrypting large volumes of data. The symmetric key is then used in the proposed approach based on DNA-based cryptographic algorithm.

Encrypt using the key after converting binary data to DNA sequencing. A DNA sequence or binary string can be the answer. Variable duration of the key. If one or both of the DNA sequence key data and DNA sequence sequences are converted to binary form, then an exclusive OR operation is performed on the corresponding DNA sequence elements and converted back to the DNA sequence. For example:

Binary form of secret data:

………… 10 10 00 01 11 10 00 1000 11 …………….

Binary form of Key:

………… 00 10 11 11 11 11 00 1110 01 …………….

XOR Result:

10 00 11 10 00 01 00 0110 10 …………….

*C. Reshaping Stage*

A simple genetic algorithm consists of three operators after encryption: replication, crossover and mutation. The reshaping process is used to generate genetic material that moves to the next activity and iteration in the form of the chromosome population. In this step, the first chromosome number and length are determined. For each round, these values can be constant and varied. Reshape it by aligning the DNA sequence into rows to create the chromosomes (chromosome population) of parents with pre-defined length.

For example:

Secret data:

………………..GCCCGCACCGGAACAACGGGCGTTCCGTCCGACCCCTTTCAACTATCAGTCTTGTCA GGCTACCGATTATCAATGCGCT

……………………….

Chromosome population

…………………….

GCCCGCACCGGAACAACGG

GCGTTCCGTCCGACCCCTTT

CAACTATCAGTCTTGTCAGG

CTACCGATTATCAATGCGCT

……………………….

1. *Crossover Stage*

The next procedure is crossover after

constructing the chromosomes of the parents.

There are two crossover forms.

These can be used sequentially in rounds of technique. In the first one, in the mating pool, parents are chosen. A single-point crossover point is then chosen between the first and last bits of the chromosomes of the parents, producing two new offspring by swapping parent 1 and parent 2 heads.

The offspring therefore contain parts of the DNA codes of both parents, as shown in the figure 1.1

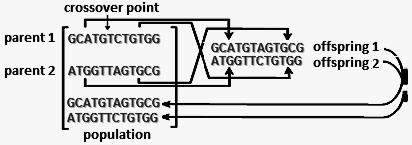


Fig.1. Two Parents Mate to produce two Offspring

Rotation is the second form of crossover, after the alignment of the Sequence of DNA into rows for chromosome construction Inhabitants. Using a predefined value to rotate left / right.

For instance:

Secret data:

………… 0 1101 1010 0101 0101 111 …………….

Apply rotate Crossover:

* 0 1010 1111 0110 1101 001 …………….

Apply one point Crossover:

Two parents:

11 0110 0100 1001 0010 11

…………….

10 0100 0111 1011 1110 01 …………….

Two offsprings:

11 0110 0100 1011 1110 01…………….

10 0100 0111 1001 0010 11…………….

*E. Mutation Stage*

The chromosomes are subject to

mutation following the crossover process.

Mutation is the modification of elements of a string. It uses two forms of mutation. Convert data to binary vector in the first one and define two mutation points between the first and last bits, then complement bits between i.e. mutation of single point changes from 1 to 0, and vice versa. In the second form of mutation, transform each of the four bits to two bases of DNA (1010  CG), for example: according to

Table 2. After conversion, convert it into a vector of DNA bases and identify two points between the first and last bases, then shift the DNA bases to each other. (i.e., CG).

Table 2. DNA bases and Representation of bits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DNA | Bits | DNA | Bits | DNA | Bits | DNA | Bits |
| TA | 0000 | GA | 0100 | CA | 1000 | AA | 1100 |
| TC | 0001 | GC | 0101 | CC | 1001 | AC | 1101 |
| TG | 0010 | GG | 0110 | CG | 1010 | AG | 1110 |
| TT | 0011 | GT | 0111 | CT | 1011 | AT | 1111 |

The following example illustrates

the proposed technique's mutation operations.

For instance:

Apply Mutation complement:

Before mutation:

* 11 0110 0100 1001 0010 11 ………….

After mutation:

………… 11 0101 1011 0110 1010 11 …………..

Apply Alter Mutation:

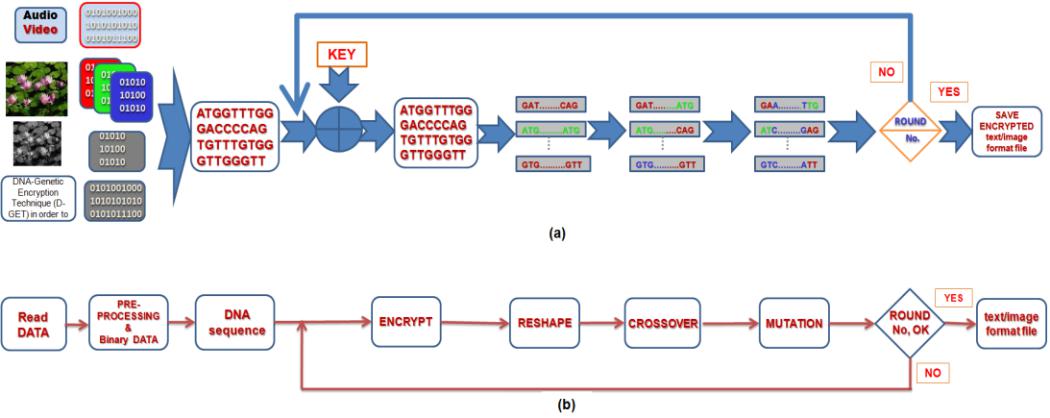
Before mutation:

…………GGACTGCGAT…………….

After mutation:

…………AAGTCATAGC…………..

The probability frequency of crossover and mutation operation is 100%. Encrypt and reshape data to pass to next round. The number of rounds depends on a predefined number of iterations. Transmit the encrypted data in text/image format file. At the receiver side, binaries received data and convert it to DNA sequencing and reshape, decrypt, crossover, mutate, decrypt and reshape to original format. The sequence of stages D-GET is illustrated in figure 2 (b). Figure 2 (a) illustrates the scenario of the stages of D-GET.



**Input:** Read Secret data (text/Image)

**Output:** encrypted (text/Image) file

BData1  Binarize the secret data.

Reshape Bdata1

Group each two adjacent bits.

DNA bases  Bdata1

While (Round No not equal to Zero) do

Encrypt Bdata2 with key.

Reshape Bdata2

Crossover operation

Mutation.

End while

Reshape Bdata2

Save encrypted (text/Image) file

Transmit file.

Fig.3. Pseudocode of Proposed Encoding D-GET

The pre-processing stage is complicated in the previous description of D-GET, meaning that for each type of data file and format, it is wide-ranging between converting

to ASCII values, reading image pixels, separating components or frames and having the video file properties as described above to solve this issue, generalising the D-GET technique. The generalisation technique consists of the same D-GET steps, but the read and pre-processing phase of the data is replaced

by a simple read binary file (fread) command using 8-bit unsigned integer (uint8) and specified parameters defining the hidden data format.

**Input**: encrypted (text/Image) file

**Output**: secret data

BData  Binarize encrypted (text/Image) file Reshape Bdata

While (Round No not equal to Zero) do

Mutation

Crossover operation

Reshape Bdata

Decrypt Bdata with key.

End while

Reshape Bdata

Save reconstracted (text/Image) file

Fig.4. Pseudocode of Proposed Decoding D-GET

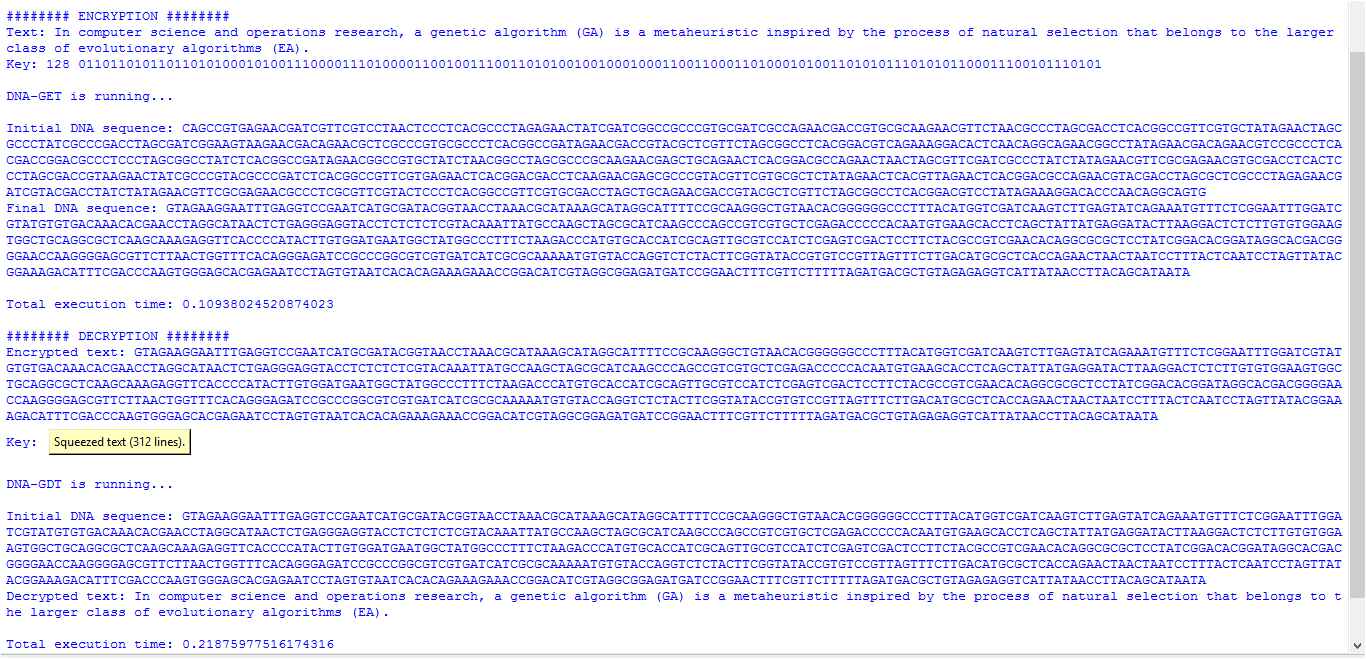
Fig.2. Stages of D-GET

The Pseudocode is shown in figure 3 and figure 4 for Encoding and Decoding secure D-GET

**IV. EVALUATION RESULTS**

The D-GET is implemented in the AMD Athlon(tm) II X2 220 Processor, 2.80GHz and 4 GB RAM on Windows 8.1 64-BIT operating system. We conduct experiments to test the efficacy of the proposed technique and run it with various types of secret data.

Using all manner of cryptanalytic, mathematical and brute-force attacks, cryptanalysts attack any encrypted data to discover its contents. A successful encryption technique against them should be robust. So, there are some features that need to be achieved. Here There is no relationship between, before encryption, sensitive data values and, after encryption, encrypted data values. Encryption should be blended around the various hidden data components so that nothing in its original position is presented.



|  |
| --- |
| **IV. EXPERIMENTAL RESULTS ANALYSIS** |
|  |

*A. Key Space Analysis*

There is a wide key space in the

D-GET that renders an attack by brute force impossible. It has the hidden keys of DNA and binary sequence forms and various combinations of them. So, it resists the exhaustive assaults of brute force.

B. Attacker Tries

Attacker attempts to discover some cypher-data information that helps him find some link between cypher-data and original data by researching how variations in an input can affect the resulting output difference in an attempt to extract the key. If the cypher-data is text, the attacker intends to make a small alteration, such as changing it; the attacker observes the plain-text change. If a hacker decodes the file, he gets another text, which further confuses him as to whether the actual information is in DNA sequence or text format. But because everything is random here, it is very difficult to decrypt. If it can not be found even with full knowledge of the decryption algorithm, the cryptosystem is safe. Experimental findings show that many current attacks can be overcome by the proposed technique. Its output is a DNA sequence that is very complex to predict the original secret. It has a greater degree of protection against some of the latest attacks.

**V.CONCLUSIONS**

D-GET is implemented in this article. The D-GET, based on multi-iteration and genetic activities, is a more stable encryption technique. Operations, encryption, rotation, crossover, mutation, and reshapes that improve the standard of encryption are also included. D-GET operations and modifications to the original data size and format. In addition, the negligible relationship in both the hidden data and its encrypted data decreases the possibilities of cryptanalysis and breaking the cypher. In addition, the technique has multilayer defence phases that achieve confidentiality and provide data with more security, productivity and robustness and protects against detection. We would standardise D-GET in future work and try to minimise transmission

**VI REFERENCES**

1. William Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, 2011.
2. Keith M. Martin,‖ Everyday Cryptography Fundamental Principles and Applications‖, Oxford University Press

Inc., New York, 2012.

1. Jingli Zheng, Zhengbing Hu and Chuiwei Lu, ―A Light-weight Symmetric Encryption Algorithm Based on Feistel

Cryptosystem Structure‖, MECS, I.J. Computer Network and Information Security, pp. 16-23, Vol. 1, 2015.

1. M.I.Khalil, ―Real-Time Encryption/Decryption of Audio Signal ―, MECS, I. J. Computer Network and Information Security, Vol., 2, pp. 25-31, 2016.
2. Mona Sabry, Mohamed Hashem and Taymoor Nazmy,

―Three Reversible Data Encoding Algorithms based on DNA and Amino Acids’ Structure‖, International Journal of Computer Applications (0975 – 8887) Volume 54– No.8, September 2012.

1. C. T. Celland, V. Risca and Bancroft C. ―Hiding messages in DNA microdots,‖ Nature, vol. 399, pp. 533–534, 1999.
2. Leier, A., Richter, C., Banzhaf, W. and Rauhe, H.,

―Cryptography with DNA Binary Strands‖, BioSystems,

Vol. 57, pp.13-22, 2000.

1. Mohammadreza, Najaftorkaman, Nazanin Sadat Kazazi, ―A Method to Encrypt Information with DNA-Based

Cryptography‖, International Journal of Cyber-Security and Digital Forensics (IJCSDF) 4(3): pp. 417-426, 2015.

1. T. Anwar, Dr. S. Paul, and S. Singh, ―Message

Transmission Based on DNA Cryptography: Review‖, International Journal of Bio-Science and Bio-Technology, Vol.6, No.5, pp.215-222, 2014.

Chen Jie, ―A DNA-based bio molecular cryptography design,‖ Proceedings of IEEE

International Symposium, Vol. 3, pp. 822-825, 2003.

[10] Noorul Hussain Ubaidur Rahman, Chithralekha

Balamurugan, and Rajapandian Mariappan, ―A Novel

DNA Computing based Encryption and Decryption

Algorithm‖, Procedia Computer Science 46 pp. 463 – 475, 2015.

1. Shima Ramesh Maniyath, Thani kaiselvan V , ―A Novel DNA based Encryption Algorithm for Multimedia information‖, An international journal of advanced computer technology, Volume 5 (1), pp. 2036 - 2045, January – 2016.
2. Pramanik Sabari, and Sanjit Kumar Setua, "DNA cryptography," In Electrical & Computer Engineering (ICECE), 7th IEEE International Conference on, pp. 551-554, 2012.
3. Tushar Mandge, Vijay Choudhary. ―A DNA encryption technique based on matrix manipulation and secure key generation scheme‖, International Conference on

Information Communication and Embedded Systems (ICICES), 21-22 Feb. 2013.

1. Sherif T. Amin, Magdy Saeb, Salah El-Gindi, "A DNA based Implementation of YAEA Encryption Algorithm," International Conference on Computational Intelligence (CI 2006), San Francisco, Nov. 20, 2006.
2. A. K. Verma, Mayank Dave, R.C. Joshi, ―Securing Ad hoc Networks Using DNA Cryptography‖, IEEE

International Conference on Computers and Devices for Communication (CODEC06), pp. 781-786, Dec. 18-20, 2006.

1. Fatma E. Ibrahim, M. I. Moussa and H. M. Abdalkader,

―A Symmetric Encryption Algorithm based on DNA Computing‖, International Journal of Computer

Applications (0975 – 8887) Volume 97– No.16, pp. 41-45, July 2014.

1. Kazuo Tanaka, Akimitsu Okamoto, and Isao Saito, ―Public-key system using DNA as a one-way function for distribution‖. Biosystems, Vol. 81, pp. 25-29, 2005.
2. Lai, XueJia, "Asymmetric encryption and signature method with DNA technology", Science China Information Sciences 53.3, pp. 506-514, 2010.
3. Cui, Guangzhao, Liming Qin, Yanfeng Wang, and Xuncai Zhang, "An encryption scheme using DNA technology", In Bio-Inspired Computing: Theories and Applications, IEEE International Conference on, pp. 37-42, 2008.
4. Lai Xin-she, Zhang Lei, ―A novel generation key scheme based on DNA‖, International conference on

Computational Intelligence and security, pp. 264-266,13-17 Dec. 2008.

**Plagiarism report**

