ASM Sc. J., 17, 2022

https://doi.org/10.32802/asmscj.2022.845

A Biological-Inspired Cryptosystem based on DNA Cryptography and Morse Code Ciphering Technique

B. Adithya[1*] and G. Santhi[2]

[1]*Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry-605014, India*

[2]*Department of Information Technology, Pondicherry Engineering College, Puducherry-605014, India*

Due to its prominent facility to hide colossal datasets with a high calibre of randomness and subsequent protection, the famous genomic Deoxyribonucleic Acid (DNA) has been presented as a hiding medium, known as DNA steganography. An incipient information-securing technique using the organic DNA structure called DNA Computing was introduced. In this paper, cryptosystem based DNA computation and Morse code cipher is proposed to bulwark the delicate knowledge within the demanding environment. The plaintext is changed over to DNA sequences utilising the encoding table. The encoded information is actually transcribed and translated by the Ribonucleic Acid (RNA) arrangements. Translated RNA is stego by the standard genetic code utilising organic compounds, and the stego DNA is ciphered by Morse code pattern. Avalanche Effect achieves 96.86% bits of average changes occurred in ciphertext. Therefore, the proposed cryptosystem shows strong randomisation and provides complex mapping between plaintext and ciphertext. Also, the designed bio explores analysis and results shows that the security of the transmission is high, and it preserves the biological process.

Keywords: Deoxyribonucleic Acid; DNA; Morse code; Cryptography; DNA cryptography; RNA

I. INTRODUCTION

For encouraging the delicate trade of data between any sender and beneficiary, safe correspondence is vital. These days, the web has become the discussion for all banking and electronic trade exchanges and it is extremely important that the connection is made in a profoundly secure way. A few strategies and frameworks have been created to scramble and unscramble the plain content in numerical cryptography to satisfy these security prerequisites. Such methodologies are anyway conquered utilising procedures and strategies for DNA cryptography. DNA cryptography is a significant control of computational DNA science. DNA cryptography plays a major part in the survival of the next generation. Numerous calculations offered in DNA cryptography have constraints, therein some of their steps either utilise standard math cryptography or upheld natural lab explores that do not appear to be suitable inside the advanced registering world.

Morse code is a text encoding system with a sequence of dashes and dots that can be transmitted over radio waves, light or sound. It can be decoded without specialised equipment by a skilled listener. In the early days, Morse code was used to apply electric telegraphs to transmit short text messages over long distance cables. During the rhythm of the Morse code, the sending operator used the Morse key (switch) to switch electrical power on and off. The electrical current engaged an electromagnet at the receiving end, which would 'press' in the Morse signals pattern. Codes were written directly on paper, in most cases by attaching the pen to an electromagnet, resulting in an initial sequence of dashes, dots, and spaces. It was perfect for broadcasting via Short Wave Radio (High Frequency), because Morse code requires a smaller bandwidth. Even if the signal was disruptive and noisy, a competent Morse operator would still be able to 'read' the text. So, the Morse code is

*Corresponding author's e-mail: adithya27.07@pec.edu

combined with DNA computing to avoid the attack played over the plaintext.

A. Requirements for DNA Computing

Each DNA computing algorithm should follow a set of specifications. In this paper, those requirements were established according to the constraints found in the current encryption algorithms as described in Table 1. Table 2 describes the execution of a series of specifications in current works of DNA computing algorithm. Following Table 2 also indicates that the accuracy of the current works in DNA cryptography specifications is not complete.

Table 1. Constraints in the current encryption algorithms

Criteria	Description
Entire character set DNA encoding	The DNA coding table must contain the complete set of characters, of which 96 elements for DNA encoding sequences.
Dynamic generation of the encoding table	At each interval session encoding table is created according to the random principle and the set of characters for each element provides concrete DNA groupings.
The sequence is unique for encoding the entire plaintext character to the sequence of DNA	At every session, each part of the plaintext character is translated into DNA sequences should be unique in each encoding table generation.
Robustness encoding	The table of character encoding must be based on a high degree of secure random encoding table must also contain random steps generations.
Preserving the biological process in the simulation	The DNA computing algorithm is focused on biological processes that are replicated to fit into the world of digital computing.
Encoding Dynamics	Due to the unique generation of DNA encoding table, different ciphertext can be generated for each session using the same plaintext

1. Entire character set DNA encoding

DNA encoding must include numbers, alphabets (upper case, lower case), and special character for entire character set fulfilment. This will be used to turn the whole set of plaintext characters into DNA groupings. The ones accessible in the current works are not finalised, and the encoding process itself does not help to enforce security. Akanksha *et al.* (2012) proposed a compatible mapping table for all characters of the numbers, alphabet (capitalised, lower case), and special character. However, it makes it very simple to decode the plaintext were the mapping table is created manually. In encoding table, the encoding sequences include all the numbers, alphabets (uppercase and lowercase), and special characters for the entire character set of 96 elements, which must be provided in order to solve this problem. Instead of replicated manual encoding table intervention for any number of times, the encoding table must be generated by a defined algorithm.

2. Dynamic generation of the encoding table

The encoding table must be re-created at regular intervals, or each session of contact between the incipient and the recipient to ensure a higher degree of protection. It's also necessary to have various DNA sequences for each aspect of the collection of characters. This objective does not fall within the limits of the current generation of algorithms for the encoding table.

3. Sequence is unique for encoding the entire plaintext character to the sequence of the DNA

For each element of the set of characters in each generation of the coding table, the encoding of the original text in the DNA sequence must be unique in each session between the

sender and the recipient. None of the current works fall under this criterion.

4. Robustness encoding

The plaintext DNA encoding is a reliable coding scheme, which is to decode, difficult to ensure resistance to attacks. Due to the manual formulation, the encoding is not stable in the current works.

5. Preserving the biological process in the simulation

In a computer environment, the DNA computing algorithm should be based on virtual biological processes that fit the digital world. Many encryption algorithms in current works are based on purely biological laboratory experiments, which are not suitable for use in the world of digital computing. In many algorithms, simulation of a biological process is taken as one part, while another part contains modern cryptography or conventional cryptography. The use of modern cryptographic algorithms or conventional cryptographic algorithms, ciphertext has been broken in DNA cryptography. Hence, the complete algorithm is important entirely on the simulation of arduous biological processes.

6. Encoding dynamics

To ensure that different ciphertexts can be produced from the same plaintext, the dynamicity of the computing method is important. That is achieved by combining traditional and DNA cryptography in the current algorithms, and not by cryptography with DNA separately (Thangavel *et al.,* 2017).

Since in the current works all of the above criteria had not been fulfilled, a novel DNA computing-based cryptosystem is proposed and depicted in Section 2. Proposed framework describes a new, effective, unique, and dynamic DNA calculation method to address this gap, and also give an overview of its results.

Table 2. Execution of a series of specifications in the current works

References	**Entire character set DNA encoding**	**Dynamic generation of the encoding table**	**Sequence is unique for encoding the entire plaintext character to the sequence of DNA**	**Robustness encoding**	**Preserving the biological process in the simulation**	**Encoding Dynamics**
Akanksha *et al.* (2012)	✓	#	#	#	#	#
Amin *et al.* (2006)	#	#	#	#	✓	#
Guangzhao *et al. (*2008)	#	#	#	#	*	*
Kang, 2008	#	#	#	#	✓	#
Mona *et al.* (2012)	#	#	#	#	*	*
Murugan and Thilagavathy (2017)	#	#	#	#	#	✓
Padma (2010)	#	#	#	#	#	#
Qiang *et al.* (2009)	#	#	#	#	*	#
Souhila *et al.* (2010)	#	#	#	#	✓	#
Tornea and Borda (2009)	#	#	#	#	*	#
Xing and Qiang (2009)	#	#	#	#	#	#
Zhang *et al.* (2010)	#	#	#	#	#	#

# Indicates minimum supporting standards * Indicates partial supporting standards ✓ Indicates Appropriate supporting standards

For the computation, this DNA cryptography uses the Central Molecular Biology Dogma (CDMB) theory. The encoding is achieved by 'short' and 'long' length of current flow through a computing within the Morse code system. The continuation of the study is structured as follows. The proposed strategy is defined in section 2. The experimental findings are discussed in Section 3. Section 3 also explains how the specifications are met. Section 4 brings this research to a conclusion.

II. MATERIALS AND METHOD

Two components of the proposed algorithm are: 1. Entire character set of DNA encoding 2. DNA computation based cryptosystem with Morse code cipher. The overall block diagram of bio-inspired cryptosystem based on DNA cryptography and Morse code cipher depicts in Figure 1.

A. Entire Character Set of DNA Encoding

The entire character set of DNA encoding is a complete description of a mechanism defined by Noorul and Chithralekha (2012). For brevity's sake, this paper focuses on the algorithm of encryption and the decryption which is maintained, and not treated in depth.

An example, the output of the algorithm for DNA encoding is shown in Table 3. However, as stated in the specifications, after each session of pre-defined intervals the DNA encoding table is created. Therefore, the sequences of DNA and the allocation of alphabet would be different between the distinctive sessions.

B. DNA Computing-based Cryptosystem with Morse Code Cipher

The computing algorithm consists of the following steps to encrypt a plaintext into ciphertext: Encoding processes to be performed before the encryption begins to transform the plaintext to Morse code.

1. Converting plaintext to DNA sequences using encoding table

Sender produces a Table 3 DNA computation for plaintext encoding using DNA encoding algorithm, and it moreover gets an index (Noorul & Chithralekha, 2012) from the recipient to build the encoding Table 3 using the same process. The plaintext encoding must be broken down into two parts.

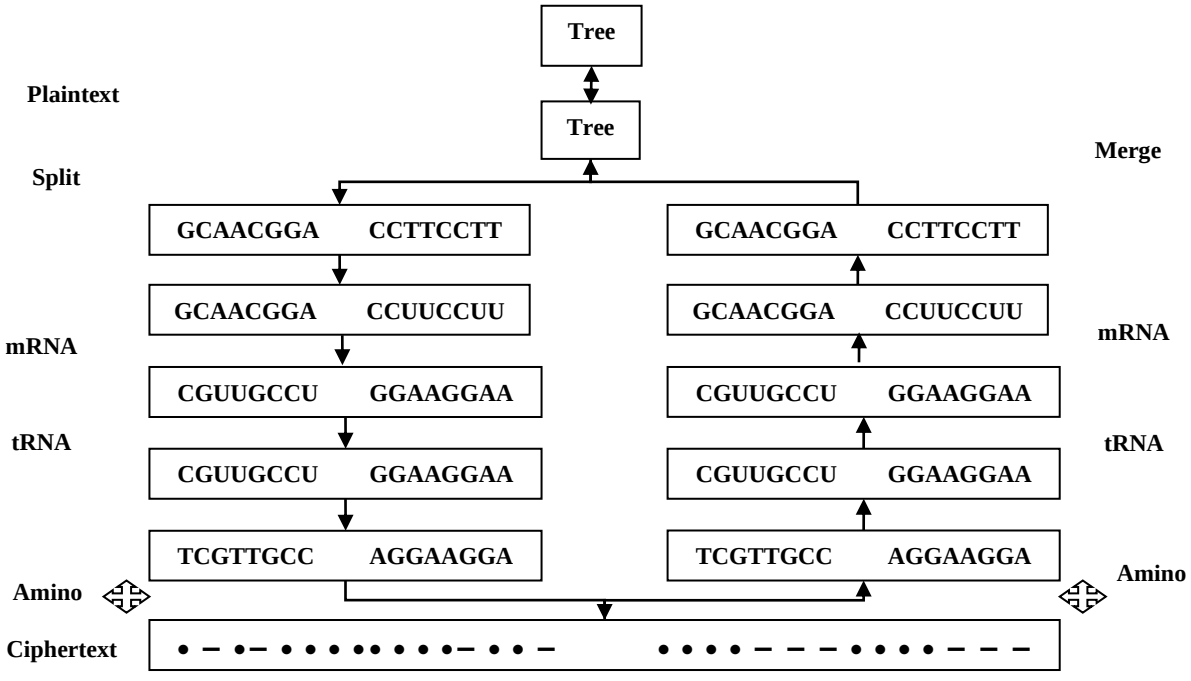


Figure 1. Block Diagram of Bio-inspired cryptosystem based on DNA cryptography and Morse code ciphering technique

Table 3. Throughput for DNA encoding

	A			T			C			G		
A	**SPACE**	-	AAGA	**8**	-	ATAA	**P**	-	ACAA	**f**	-	AGCA
	!	-	AACG	**9**	-	ATGG	**Q**	-	ACGG	**g**	-	AGAG
	“	-	AATT	**:**	-	ATTT	**R**	-	ACCT	**h**	-	AGCT
	#	-	AAAC	**;**	-	ATCC	**S**	-	ACTC	**i**	-	AGCC
G	**$**	-	GAGA	**<**	-	GTGA	**T**	-	GCAA	**j**	-	GGAA
	%	-	GACG	**=**	-	GTCG	**U**	-	GCGG	**k**	-	GGGG
	&	-	GACT	**>**	-	GTTT	**V**	-	GCTT	**l**	-	GGCT
	‘	-	GAGC	**?**	-	GTAC	**W**	-	GCCC	**m**	-	GGTC
T	**(**	-	TACA	**@**	-	TTGA	**x**	-	TCGA	**n**	-	TGAA
	)	-	TAAG	**A**	-	TTCG	**Y**	-	TCCG	**o**	-	TGGG
	*	-	TACT	**B**	-	TTCT	**Z**	-	TCTT	**p**	-	TGTT
	+	-	TACC	**C**	-	TTGC	**[**	-	TCAC	**q**	-	TGCC
C	**,**	-	CAAA	**D**	-	CTCA	\	-	CCGA	**r**	-	CGGA
	-	-	CAGG	**E**	-	CTAG	**]**	-	CCCG	**s**	-	CGCG
	.	-	CACT	**F**	-	CTCT	^	-	CCCT	**t**	-	CGTT
	/	-	CATC	**G**	-	CTCC	_	-	CCGC	**u**	-	CGAC
A	**0**	-	AATG	**H**	-	ATAG	`	-	ACAG	**v**	-	AGGG
	1	-	AACT	**I**	-	ATCT	~	-	ACTT	**w**	-	AGTT
G	**2**	-	GATG	**J**	-	GTTG	**€**	-	GCAG	**x**	-	GGAG
	3	-	GATT	**K**	-	GTCT	**a**	-	GCCT	**y**	-	GGTT
T	**4**	-	TAGG	**L**	-	TTTG	**b**	-	TCTG	**z**	-	TGAG
	5	-	TATT	**M**	-	TTTT	**c**	-	TCCT	**{**	-	TGCT
C	**6**	-	CAAG	**N**	-	CTGG	**d**	-	CCTG	|	-	CGTG
	7	-	CATT	**O**	-	CTTT	**e**	-	CCTT	**}**	-	CGCT

In case the plaintext is still not equally partitioned, an arbitrary variable is added to make it equal. Using the sender-accessible DNA encoding Table 3, one part of the plaintext is transformed to DNA sequence and using the DNA encoding receiver Table 3, the other part of the plaintext is transformed to DNA sequence. For illustration: Let’s expect the plaintext is, “DNA Ok.” The plaintext is equally separated into two parts, and let’s expect the results of DNA are the encoding of Table 3 as follows:

DNA	SPACEOK
CTCA CTGG TTCG	**AAGA CTTT GTCT**

2. Transcription and translation process of obtained DNA sequences

Multiple round functions are performed in both left and right sequences, and the following are multi-round feature steps: Minimum number of rounds must be equal to or greater than 10. Normally, plaintext is first changed over to DNA sequence by utilising the following binary code rule A-00, T-01, C-10, G-11 (Adithya & Santhi, 2020). Respectively, intron sequences are used to generate the sender and the receiver encoding Table 3. These two intron sequences are used simultaneously for plaintext sequence coding to carry out transformation operations for both the right and left side components. The transformation process is essentially a plaintext encoded with DNA, with the respective sequences of intron. The transformed DNA sequence is converted into a messenger Ribonucleic Acid (mRNA) sequence on the left and right sides of the DNA sequence after the substitution of Thymine (T) with Uracil (U). It is a tool for simulating a system for biological transcriptions. For instance:

CTCA CTGG TTCG	**AAGA CTTT GTCT**
Transcription	
CUCA CUGG UUCG	**AAGA CUUU GUCU**

The complementary DNA alphabet is applied over each DNA alphabet to translate mRNA sequence into transfer Ribonucleic Acid (tRNA) sequence. Conversions have to be rendered to A-U, U-A, G-C, C-G. This procedure is a biological method of simulation translation. For instance:

CUCA CUGG UUCG	**AAGA CUUU GUCU**
Translation	
GAGU GACC AAGC	**UUCU GAAA CAGA**

A substitution of Uracil (U) with Thymine (T), the tRNA sequence is transformed to DNA sequence. This is an organic method of reverse-transcription of the simulation. The reverse-transcription DNA sequence is moved once on the two sides of the right-hand side. For example:

GAGT GACC AAGC	TTCT GAAA CAGA
Shifting	
CGAG TGAC CAAG	TTCT GAAA CAGA

3. Generation of amino acid encoding table

The obtained tRNA sequence must be taken and transformed to amino acids after several trials. For that analysis, a corresponding sequence of amino acids is needed for every tRNA chain. This study results in an adequate amino acid table. Randomly produce two DNA sequences with four DNA alphabets. All four chemical compounds from the DNA should be the creation of sequences; the sequences should not be identical and is shown in Table 4.

Table 4. Generated amino acid encoding table

	T	**G**	**A**	**C**
A	TA	GA	AA	CA
G	TG	GG	AG	CG
T	TT	GT	AT	CT
C	TC	GC	AC	CC

The universal basic amino acids table consists of 20 amino acids. Table 6 (a,b) indicates the 256 amino acid elements which are extended from 20 amino acids. Group partitioning with respective to DNA nucleotide (Wikipedia amino acid, 2019) are categorised as four classes: A, T, C and G is shown in Table 5. Such 256 elements are individually allotted with one group partitioning is shown in Table 7 (a,b,c,d) where Elem- Elements. For example:

CGAG TGAC CAAG			TTCT GAAA CAGA		
Encoded stego DNA Sequence					
Y7	**H6**	**AU**	**Q7**	**V7**	**V2**

Table 5. Group partitioning with respective to DNA nucleotide

DNA Nucleotide	**Partitioned Group**
A	A0, A1, A2, A3, A4, A5, A6, A7, AU, AO, AX, AB, AZ, AJ, R0, R1, R2, R3, R4, R5, R6, R7, RU, RO, RX, RB, RZ, RJ, N0, N1, N2, N3, N4, N5, N6, N7, NU, NO, NX, NB, NZ, NJ, D0, D1, D2, D3, D4, D5, D6, D7, DU, DO, DX, DB, DZ, DJ, C0, C1, C2, C3, C4, C5, C6, C7, CU, CO, CX, CB, CZ, CJ
T	L0, L1, L2, L3, L4, L5, L6, L7, LU, LO, LX, LB, LZ, LJ, K0, K1, K2 K3, K4 K5 K6, K7, KU, KO, KX, KB, KZ, KJ, M0, M1, M2, M3, M4, M5, M6, M7, MU, MO, MX, MB, MZ, MJ, F0, F1, F2 F3, F4, F5, F6, F7, FU, FO, FX, FB, FZ, FJ, P0, P1, P2, P3, P4, P5, P6, P7, PU, PO, PX, PB, PZ, PJ
G	Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, QU, QO, QX, QB, QZ, QJ, E0, E1, E2, E3, E4, E5, E6, E7, EU EO, EX, EB, EZ, EJ, G0, G1, G2, G3, G4, G5, G6 G7, GU, GO, GX, GB GZ, GJ, H0, H1, H2, H3, H4, H5, H6, H7, HU, HO, HX, HB, HZ, HJ, I0, I1, I2, I3, I4, I5, I6, I7, IU, IO, IX IB, IZ, IJ
C	S0, S1, S2, S3, S4, S5, S6, S7, SU, SO, SX, SB, SZ, SJ, T0, T1, T2, T3, T4, T5, T6, T7, TU, TO, TX, TB, TZ, TJ, W0, W1, W2, W3, W4, W5, W6, W7, WU, WO, WX, WB, WZ, WJ, Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, YU, YO, YX, YB, YZ, YJ

Table 6. Extended amino acid encoding table with 256 elements

(a)

	TA	**GA**	**AA**	**CA**	**TG**	**GG**	**AG**	**CG**
TC	TCTA	TCGA	TCAA	TCCA	TCTG	TCGG	TCAG	TCCG
AC	ACTA	ACGA	ACAA	ACCA	ACTG	ACGG	ACAG	ACCG
GC	GCTA	GCGA	GCAA	GCCA	GCTG	GCGG	GCAG	GCCG
CC	CCTA	CCGA	CCAA	CCCA	CCTG	CCGG	CCAG	CCCG
CT	CTTA	CTGA	CTAA	CTCA	CTTG	CTGG	CTAG	CTCG
GT	GTTA	GTGA	GTAA	GTCA	GTTG	GTGG	GTAG	GTCG
AT	ATTA	ATGA	ATAA	ATCA	ATTG	ATCG	ATAG	ATCG
TT	TTTA	TTGA	TTAA	TTCA	TTTG	TTGG	TTAG	TTCG
TG	TGTA	TGGA	TGAA	TGCA	TGTG	TGGG	TGAG	TGCG
AG	AGTA	AGGA	AGAA	AGCA	AGTG	AGGG	AGAG	AGCG
GG	GGTA	GGGA	GGAA	GGCA	GGTG	GGGG	GGAG	GGCG
CG	CGTA	CGGA	CGAA	CGCA	CGTG	CGGG	CGAG	CGCG
CA	CATA	CAGA	CAAA	CACA	CATG	CAGG	CAAG	CACG
GA	GATA	GAGA	GAAA	GACA	GATG	GAGG	GAAG	GACG
AA	AATA	AAGA	AAAA	AACA	AATG	AAGG	AAAG	AACG
TA	TATA	TAGA	TAAA	TACA	TATG	TAGG	TAAG	TACG

(b)

	TT	**GT**	**AT**	**CT**	**TC**	**GC**	**AC**	**CC**
TC	TCTT	TCGT	TCAT	TCCT	TCTC	TCGC	TCAC	TCCC
AC	ACTT	ACGT	ACAT	ACCT	ACTC	ACGC	ACAC	ACCC
GC	GCTT	GCGT	GCAT	GCCT	GCTC	GCGC	GCAC	GCCC
CC	CCTT	CCGT	CCAT	CCCT	CCTC	CCGC	CCAC	CCCC
CT	CTTT	CTGT	CTAT	CTCT	CTTC	CTGC	CTAC	CTCC
GT	GTTT	GTGT	GTAT	GTCT	GTTC	GTGC	GTAC	GTCC
AT	ATTT	ATGT	ATAT	ATCT	ATTC	ATGC	ATAC	ATCC
TT	TTTT	TTGT	TTAT	TTCT	TTTC	TTGC	TTAC	TTCC
TG	TGTT	TGGT	TGAT	TGCT	TGTC	TGGC	TGAC	TGCC
AG	AGTT	AGGT	AGAT	AGCT	AGTC	AGGC	AGAC	AGCC
GG	GGTT	GGGT	GGAT	GGCT	GGTC	GGGC	GGAC	GGCC
CG	CGTT	CGGT	CGAT	CGCT	CGTC	CGGC	CGAC	CGCC
CA	CATT	CAGT	CAAT	CACT	CATC	CAGC	CAAC	CACC
GA	GATT	GAGT	GAAT	GACT	GATC	GAGC	GAAC	GACC
AA	AATT	AAGT	AAAT	AACT	AATC	AAGC	AAAC	AACC
TA	TATT	TAGT	TAAT	TACT	TATC	TAGC	TAAC	TACC

Table 7. Extended amino acid table with group partitioning

(a)

	TA		**GA**		**AA**		**CA**	
	Elem	**Group**	**Elem**	**Group**	**Elem**	**Group**	**Elem**	**Group**
TC	TCTA	K0	TCGA	K4	TCAA	KU	TCCA	KZ
AC	ACTA	K1	ACGA	K5	ACAA	KO	ACCA	KJ
GC	GCTA	K2	GCGA	K6	GCAA	KX	GCCA	M0
CC	CCTA	K3	CCGA	K7	CCAA	KB	CCCA	M1
CT	CTTA	F4	CTGA	FU	CTAA	FZ	CTCA	P2
GT	GTTA	F5	GTGA	FO	GTAA	FJ	GTCA	P3
AT	ATTA	F6	ATGA	FX	ATAA	P0	ATCA	P4
TT	TTTA	F7	TTGA	FB	TTAA	P1	TTCA	P5
TG	TGTA	SU	TGGA	SZ	TGAA	T2	TGCA	T6

AG	AGTA	SO	AGGA	SJ	AGAA	T3	AGCA	T7
GG	GGTA	SX	GGGA	T0	GGAA	T4	GGCA	TU
CG	CGTA	SB	CGGA	T1	CGAA	T5	CGCA	TO
CA	CATA	YZ	CAGA	V2	CAAA	V6	CACA	VX
GA	GATA	YJ	GAGA	V3	GAAA	V7	GACA	VB
AA	AATA	V0	AAGA	V4	AAAA	VU	AACA	VZ
TA	TATA	V1	TAGA	V5	TAAA	VO	TACA	VJ

(b)

	TG		**GG**		**AG**		**CG**	
	Elem	**Group**	**Elem**	**Group**	**Elem**	**Group**	**Elem**	**Group**
TC	TCTG	M2	TCGG	M6	TCAG	MX	TCCG	F0
AC	ACTG	M3	ACGG	M7	ACAG	MB	ACCG	F1
GC	GCTG	M4	GCGG	MU	GCAG	MZ	GCCG	F2
CC	CCTG	M5	CCGG	MO	CCAG	MJ	CCCG	F3
CT	CTTG	P6	CTGG	PX	CTAG	S0	CTCG	S4
GT	GTTG	P7	GTGG	PB	GTAG	S1	GTCG	S5
AT	ATTG	PU	ATGG	PZ	ATAG	S2	ATCG	S6
TT	TTTG	PO	TTGG	PJ	TTAG	S3	TTCG	S7
TG	TGTG	TX	TGGG	Y0	TGAG	Y4	TGCG	YU
AG	AGTG	TB	AGGG	Y1	AGAG	Y5	AGCG	YO
GG	GGTG	TZ	GGGG	Y2	GGAG	Y6	GGCG	YX
CG	CGTG	TJ	CGGG	Y3	CGAG	Y7	CGCG	YB
CA	CATG	A0	CAGG	A4	CAAG	AU	CACG	AZ
GA	GATG	A1	GAGG	A5	GAAG	AO	GACG	AJ
AA	AATG	A2	AAGG	A6	AAAG	AX	AACG	R0
TA	TATG	A3	TAGG	A7	TAAG	AB	TACG	R1

(c)

	TT		**GT**		**AT**		**CT**	
	Elem	**Group**	**Elem**	**Group**	**Elem**	**Group**	**Elem**	**Group**
TC	TCTT	R2	TCGT	R6	TCAT	RX	TCCT	N0
AC	ACTT	R3	ACGT	R7	ACAT	RB	ACCT	N1
GC	GCTT	R4	GCGT	RU	GCAT	RZ	GCCT	N2
CC	CCTT	R5	CCGT	RO	CCAT	RJ	CCCT	N3
CT	CTTT	D6	CTGT	DX	CTAT	Q0	CTCT	Q4
GT	GTTT	D7	GTGT	DB	GTAT	Q1	GTCT	Q5
AT	ATTT	DU	ATGT	DZ	ATAT	Q2	ATCT	Q6
TT	TTTT	DO	TTGT	DJ	TTAT	Q3	TTCT	Q7
TG	TGTT	EX	TGGT	G0	TGAT	G4	TGCT	GU
AG	AGTT	EB	AGGT	G1	AGAT	G5	AGCT	GO
GG	GGTT	EZ	GGGT	G2	GGAT	G6	GGCT	GX
CG	CGTT	EJ	CGGT	G3	CGAT	G7	CGCT	GB
CA	CATT	L0	CAGT	L4	CAAT	WU	CACT	WZ
GA	GATT	L1	GAGT	L5	GAAT	WO	GACT	WJ
AA	AATT	L2	AAGT	L6	AAAT	WX	AACT	C0
TA	TATT	L3	TAGT	L7	TAAT	WB	TACT	C1

(d)

	TC		**GC**		**AC**		**CC**	
	Elem	**Group**	**Elem**	**Group**	**Elem**	**Group**	**Elem**	**Group**
TC	TCTC	N4	TCGC	NU	TCAC	NZ	TCCC	D2
AC	ACTC	N5	ACGC	NO	ACAC	NJ	ACCC	D3

GC	GCTC	N6	GCGC	NX	GCAC	D0	GCCC	D4
CC	CCTC	N7	CCGC	NB	CCAC	D1	CCCC	D5
CT	CTTC	QU	CTGC	QZ	CTAC	E2	CTCC	E6
GT	GTTC	QO	GTGC	QJ	GTAC	E3	GTCC	E7
AT	ATTC	QX	ATGC	E0	ATAC	E4	ATCC	EU
TT	TTTC	QB	TTGC	E1	TTAC	E5	TTCC	EO
TG	TGTC	GZ	TGGC	H2	TGAC	H6	TGCC	HX
AG	AGTC	GJ	AGGC	H3	AGAC	H7	AGCC	HB
GG	GGTC	H0	GGGC	H4	GGAC	HU	GGCC	HZ
CG	CGTC	H1	CGGC	H5	CGAC	HO	CGCC	HJ
CA	CATC	C2	CAGC	I6	CAAC	IX	CACC	W0
GA	GATC	C3	GAGC	I7	GAAC	IB	GACC	W1
AA	AATC	C4	AAGC	IU	AAAC	IZ	AACC	W2
TA	TATC	C5	TAGC	IO	TAAC	IJ	TACC	W3

4. Applying Morse code pattern to generate ciphertext

During the 1830s, American artist and inventor Samuel F.B. Morse created the method for electrical telegraphy in the United States (Pathak, 2020). Morse's helper and colleague, American scientist and businessman Alfred Lewis Vail, enhanced this version much further (Morse Code, 2019). Morse code is a way to transfer information by using the regular sequence of short and long marks or pulses of letters, numbers and special characters of a message, usually referred to as "dots and dashes" appears in Table 8. The Morse code system's brilliance is that it can also be transmitted as an audio signal, and was originally used to send messages over the telegraph (before inventing telephones). In this form, a short beep is given to each dot and a dash to a longer beep (three times the length of a dot).

Recalling the Morse codes is an art of its own. In our mind, it means a rhythmic response. To this end, a dot (.) is called as 'di' and a dash (-) is called as 'dah'[1]. A 'di' is pronounced at the end of the combination as 'dit'. A simple example of Morse code is '**. - .**' (di dah dit) for letter R. A message that includes 125 letters will make the speed 5 words per minute (wpm) when sent in 5 minutes or when received in 5 minutes. 125 divided by 5 gives 25 words in 5 minutes, i.e., 5 wpm. So, the minimum speed to transfer 125 letters is 5 wpm[2]. The Morse code pattern is applied over the encoded DNA sequences to form the compressed ciphertext. For example:

Y7	**H6**	**AU**	**Q7**	**V7**	**V2**
Ciphertext					
– • – – – – • • •	• • • • – • • • •	• –• • –	– – • – – – • • •	• • • – – – • • •	• • • –• • – – –

5. Extraction of plaintext from the Morse cipher

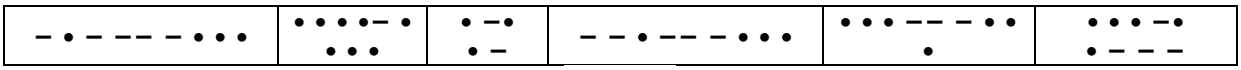
From the secure connection, the receiver receives the encrypted message with the concept of the intermediate network, and the recipient employments the DNA encoding algorithm[3] to create two DNA encoding tables from their claim hint and the sender hint. By using the amino acid table given in Table 7, the tRNA sequence is created by translating the protein groupings on both the left and right sides of the sequences. The DNA sequences in left and right sides are moved one time to the left. Uracil (U) converts the left as well as the right side of the transferred sequence into mRNA as a Thymine (T). This is how the reverse simulation is biologically transcribed. The complementary DNA alphabet is applied over each DNA alphabet to translate the mRNA sequences to tRNA sequences. Illustration analyses include A-U, U-A, G-C, C-G. This is the biological simulation form of reverse translation.

[1]https://www.cryptomuseum.com/radio/morse/
[2]https://vigyanprasar.gov.in/learning-the-morse-code/
[3]https://www.tutorialspoint.com/cryptography/attacks_on_cryptosystems.htm

Table 8. Morse code pattern[4]

Characters	Morse code pattern	Characters	Morse code pattern
A	• –	S	• • •
B	– • • •	T	–
C	– • – •	U	• • –
D	– • •	V	• • • –
E	•	W	• – –
F	• • – •	X	– • • –
G	– – •	Y	– • – –
H	• • • •	Z	– – • •
I	• •	0	– – – – –
J	• – – –	1	• – – – –
K	– • –	2	• • – – –
L	• – • •	3	• • • – –
M	– –	4	• • • • –
N	– •	5	• • • • •
O	– – –	6	– • • • •
P	• – – •	7	– – • • •
Q	– – • –	8	– – – • •
R	• – •	9	– – – – •

The tRNA groupings is translated to DNA sequence on both right and left by substituting Uracil (U) with Thymine (T). The translation of the DNA sequence in the right and left is translated into plaintext by means of DNA decoding in Table 3. Finally, both the left and the right side plaintext are blended. For example:



DNA **Plaintext** **SPACEOK**

III. RESULT AND DISCUSSION

The purpose of the attacker is to crack the cryptosystem and extract plain text from the text of the cipher. The attacker just needs to find the decryption key secret to get the plaintext, as the algorithm already lies in the public domain[3]. Once the attacker can determine the key, the attacked system will be deemed to be broken or damaged (Aich *et al.,* 2015). Attacks on encryption systems are categorised based on the proposed cryptographic algorithm as Brute-Force Attack, Ciphertext-Only Attack, Linear Cryptanalysis Attack, Differential Cryptanalysis Attack, and Timing Attacks. Avalanche effect, confusion and diffusion in cryptography are the properties of stable cipher operation that were defined by Claude[5] as follows.

A. Confusion and Diffusion

In confusion, the key is not directly related with the ciphertext in a simple way. Every character of the encrypted text would depend upon various parts of the key. In proposed cryptosystem, amino acid table contains $n \times n$ matrix and a pair of length is n^2 to solve the encryption process. If one of the ciphertext characters changes, a whole column of the matrix will change completely. Simultaneously, cryptanalysts would probably need to resolve the key of the whole instead of piece by piece process.

[4]http://ascii-table.com/morse-code.php
[5]https://crypto.interactive-maths.com/other-examples1.html

In diffusion, character of the plaintext changes then ciphertext character will also be changed immediately and vice versa. In proposed cryptosystem, several characters in the plaintext are diffused over ciphertext. The plaintext or the key cannot be used independently to determine the corresponding encrypted text in which multiple operations are performed to combine the incipient and the receiver to produce the result. When a single bit of plaintext is changed, the transcription and the translation process will produce changes within the encrypted text. It is too critical to note that different ciphertexts will be created using the proposed cryptosystem on the same plaintext since the key will be created as distinctive for each session. Finally, the proposed cryptosystem achieves the confusion and diffusion properties.

B. Avalanche Effect

It is the property of cryptosystem where a small change in the key or the plaintext ought to be resulted with a significant change in the ciphertext. In proposed framework, one bit of the encoding plaintext changes randomly which results in 94.12% bits of the average changes occurred in ciphertext. A two bit of the encoding plaintext changes randomly which results in 96.86% bits of the average changes occurred in ciphertext. This process shows strong randomisation of the proposed cryptosystem and provides a complex mapping between plaintext and ciphertext.

C. Security against Attacks

1. Brute-Force Attack (BFA)

A Brute-Force Attack is a method of decoding sensitive data using trial-and-error. Password cracking and encryption key cracking are the most common methods for brute force attacks (OWASP, 2017). In proposed cryptosystem, the different combination of encoding table will be tried until the correct combination of the encoding table is found with the Morse code pattern. Encoding table contains $n \times n$ matrix, n Morse pattern and a pair of length is $n * n^2$ to find the ciphertext. Even one table of the Morse pattern is guessed, it is difficult to find other encoding patterns, i.e., generated encoding table for each character and amino acid encoding table for partitioned group need to be derived to extract the ciphertext. Moreover, without any indicated hashed microdot values, it is computationally difficult to map all conceivable microdot outcomes to their corresponding encoding. Therefore, the proposed cryptosystem for the Brute-Force Attack is secure.

2. Ciphertext-Only Attack (COA)

Ciphertext-Only Attacks are said to be successful or effective when the relevant plaintext is evaluated from a particular set of ciphertexts. Also via this attack the encryption key can be determined. This did not work in the proposed system, due to the high entropy of the generated key to encode the plaintext with specific pattern generation. Further operations contribute to greater plaintext obscurity. In other words, the more frequent plaintext character does not map to the ciphertext character that occurs the most often. Due to the distinctive sizes of plaintext and ciphertext (as observed in all re-created test cases), the relation between plaintext and ciphertext becomes more complicated. Therefore, the proposed cryptosystem is safe from Ciphertext-Only Attack.

3. Linear Cryptanalysis Attack (LCA)

In Linear Cryptanalysis Attack, for certain portions of the ciphertext, the attacker knows the plaintext. The job is to use the data to decipher the rest of the ciphertext. It can be achieved by defining the goal, or using a variety of other techniques. Decoding the key as DNA computing where it includes a lot of large-scale computations is troublesome. In this context, even though the attacker recovers the key from the table set, for every execution of the method a distinctive set of keys is crisply generated. Proposed cryptanalysis can thus withstand the attack of the Linear Cryptanalysis.

4. Differential Cryptanalysis Attack (DCA)

Differential cryptanalysis attack may be a selected plaintext attack in which the attacker would infer ciphertexts from a collection of selected plaintext. Because of the fresh era of keys for each session, this attack also fails which results in completely different ciphertexts.

5. Timing Attack (TA)

It is a fact that the distinction between the calculations take different times to compute on the processor. By measuring such timings, it is conceivable that the processor's performing almost a specific computation. In proposed strategy, without any space the Morse pattern is created as ciphertext.

Even though the Morse experts can able to find the Morse patterns, they take billions chain map to scramble the encoding table of DNA computing as well as amino acid encoding table. Therefore, the proposed encryption of the cryptographic system takes longer. It shows that the secret key is long. Table 9 summarises the comparison of the proposed framework relating to specific attacks.

Table 9. Comparison of proposed framework relating to specific attacks

References	Brute-Force Attack	Ciphertext-only Attack	Linear Cryptanalysis Attack	Differential Cryptanalysis Attack	Timing Attack
Kang, 2008	✓	✓	-	-	-
Padma, 2010	✓	-	-	-	-
Souhila *et al.*, 2010	✓	#	✓	✓	#
Murugan and Thilagavathy, 2017	✓	✓	-	✓	#
Proposed DNA-Morse code strategy	✓	✓	✓	✓	✓

✓ Indicates prevention of attack # Indicates failure to prevent the attack

D. CIA Triad

CIA stands for confidentiality, integrity and availability, otherwise known as the CIA Triad. Together, these three principles are the cornerstones of every security infrastructure; in fact, they work as the goals and objectives of each program of security.

1. Confidentiality (C)

Confidentiality implies the detention secret from unauthorised entities during the exchange of information. This has been created in the proposed system, as it scrambled all the transferred entities and parameters.

2. Integrity (I)

Data integrity ensures there is no manipulated information is obtained from the beneficiary as a result of insertion, deletion and modifications. In other words, if the data is altered, the receiver will have some component frame to ensure that the information obtained is up-to-date. In this conspire, data integrity was achieved through the use of encoding table. Unless the attacker maliciously alters the information before the receiver receives it and the microdot values may differ between the incipient and the recipient.

3. Availability (A)

The availability involves ensuring that individuals have access to data whenever appropriate (Adithya and Santhi, 2021). The proposed cryptosystem can be a virtual one and, therefore accessible at each session. It is also supported massive types of content input to the encryption and decryption of data without loss of data. i.e., it is a loss-free system, where no bit is lost between the transmissions. The proposed cryptosystem algorithm was also evaluated for sizes up to 10,000 plaintext characters with spaces and special characters.

E. Fulfilment of DNA Computing Requirements

It is evident in this algorithm that the stipulated criteria are fulfilled entirely is shown in Table 10.

Table 10. Stipulated criteria's are fulfilled entirely

Criteria	Description
Entire character set DNA encoding	For all the set of characters, the encoding sequence is listed in encoding table is seen in Table 5.
Dynamic generation of the encoding table	After each session of the interval between the incipient and the recipient, the encoding table is created at random.
The sequence is unique for encoding the entire plaintext character to the sequence of DNA	Plaintext encoding in DNA sequence in each encoding table generation in each session is unique for each character of the element.
Robustness encoding	Because of more randomness, the robust encoding scheme is given.
Preserving the biological process in the simulation	The DNA computing algorithm is based on the CDMB simulated translation and transcription processes to adapt to the current computing environment.
Encoding Dynamics	It fulfils the necessity. Due to the specific generation of the DNA encoding table, different ciphertext for each session will be generated in the same plaintext.

IV. CONCLUSION

A bio-inspired cryptosystem that proposes the DNA computing with Morse code encoding to secure the data during transmission. In this strategy, the encoding table is created for alpha-numeric letter set. Plaintext is first converted into DNA sequences as per the encoding table. The transcription process was carried out to turn the DNA sequences into mRNA sequences on both left and right side DNA groupings by swapping Thymine (T) with Uracil (U). The mRNA arrangement is transformed into tRNA grouping to process the biological translation by swapping each set of DNA letters with its complement DNA alphabet. Every grouping of tRNA includes an adequate amino acid sequence, and a suitable amino acid table is established. The 20 amino acids would comprise 256 components and that are partitioned into four groups such as A, C, G, and T. Now, Morse code pattern is applied over the encoded DNA sequences to form the ciphertext. Avalanche effect shows that two bit of the encoding plaintext changes randomly which results in 96.86% bits of the average changes occurred in ciphertext. The proposed framework has been proven to be secure from BFA, COA, LCA, DCA, and TA. Moreover, DNA computing based cryptosystem with Morse pattern achieves the CIA principles. A characteristic of proposed algorithm is a unique biological simulation that fulfils all the functional and non-functional attributes.

Future work in this area will require new security and authentication to strengthen the mechanisms of the scheme. Artificial Intelligence also has wide possibilities in the field of cryptography. DNA cryptography is still in its infancy and this computational paradigm can be used to construct robust cryptosystem.

V. REFERENCES

Adithya, B & Santhi, G 2020, 'Bio-inspired Deoxyribonucleic Acid based data obnubilating using Enhanced Computational Algorithms', in: Proceedings of the International Conference on Computer Networks, Big Data and IoT, Springer, pp. 597-609.

Adithya, B & Santhi, G 2021, 'DNA computing using cryptographic and steganographic strategies', Data Integrity and Quality, pp. 1-19.

Aich, Sen, Dash & Dehuri 2015, 'A symmetric key cryptosystem using DNA Sequence with OTP key', Advances in Intelligent Systems and Computing, pp. 207-215.

Akanksha, A, Akansha, B, Jaya, S, Meer, SA & Divya, G 2012, 'Implementation of DNA algorithm for secure voice communication', International Journal of Scientific & Engineering Research, vol. 3, no. 6, pp. 1-5.

Amin, ST, Magdy, S & Salah, EG 2006, 'A DNA-based implementation of YAEA encryption algorithm', in: Proceedings of the International Conference on Computational Intelligence, IASTED, pp. 120-125.

Guangzhao, C, Limin, Q, Yanfeng, W & Xuncai, Z 2008, 'An encryption scheme using DNA technology', in: Proceedings of the IEEE 3rd International Conference on Bio-Inspired Computing: Theories and Applications, United States, pp. 37-42.

Kang, N 2009, 'A pseudo DNA cryptography method', viewed 27 May 2019, <http://arxiv.org/abs/0903.269>.

Mona, S, Mohamed, H & Taymoor, N 2012, 'Three reversible data encoding algorithms based on DNA and amino acids structure', International Journal of Computer Applications, vol. 54, no. 8.

Morse Code viewed 23 July 2019, <https://www.britannica.com/topic/Morse-Code>.

Murugan, A & Thilagavathy, R 2017, 'Securing cloud data using DNA and Morse code: A triple encryption scheme', International Journal of Control Theory and Applications, vol. 10, no. 23, pp. 31-38.

Noorul, HU & Chithralekha, T 2012, 'A novel DNA encoding technique and system for DNA cryptography', India Patent 5107, CHE.

OWASP, Brute Force Attack, viewed 23 May 2017, <https://www.owasp.org/index.php/Brute_force_attack> .

Padma, B 2010, DNA computing theory with ECC, viewed 9 October 2018, <http://www.scribd.com/doc/55154238/Report>.

Pathak, P 2020, 'History of Morse Code', International Journal of Research: Education, <https://internationaljournalofresearch.com/2020/07/15/history-of-morse-code-2/>.

Qiang, Z, Ling, G, Xianglian, X & Xiaopeng, W 2009, 'An image encryption algorithm based on DNA sequence addition operation', in: Proceedings of IEEE 4th International Conference on Bio-Inspired Computing: Theories and Applications, pp. 16-19.

Souhila, S, Mohamed, G, Mansouri, N & Drias, H 2010, 'An encryption algorithm inspired from DNA', in: Proceedings of the IEEE International Conference on Machine and Web Intelligence, pp. 344-349.

Thangavel, M, Varalakshmi, P & Sindhuja, R 2017, 'A comparative study on DNA-Based cryptosystem', Handbook of Research on Recent Developments in Intelligent Communication Application, pp. 496-528.

Tornea, O & Borda, ME 2009, 'DNA Cryptographic Algorithms', in: IFMBE Proceedings of the International Conference on Advancements of Medicine and Health Care through Technology, Springer, Romania, pp. 223-226.

Wikipedia amino acid, viewed 27 July 2019, <https://en.wikipedia.org/wiki/Amino_acid>.

Xing, W & Qiang, Z 2009, 'DNA computing-based cryptography', IEEE Fourth International on Conference on Bio-Inspired Computing, pp. 67-69.

Zhang, Qiang, Wang, Qian, Wei & Xiaopeng 2010, 'A novel image encryption scheme based on DNA coding and multi-chaotic map', Advanced Science Letters, vol. 3, pp. 447-451.