ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.10

A DNA and Amino Acids-Based Implementation of Playfair Cipher

Geetanjali sharma[1], Om Prakash Pal[2]
M.Tech Scholar[1], Assistance Professor[2]
Department of Computer Science and Engineering
Bharat institute of technology, Meerut, India

Abstract:
The DNA cryptography is a new and very promising direction in cryptography research. Although in its primitive stage, DNA cryptography is shown to be very effective. Currently, several DNA computing algorithms are proposed for quite some cryptography, cryptanalysis and steganography problems, and they are very powerful in these areas. This paper discusses a significant modification to the old Playfair cipher by introducing DNA-based and amino acids-based structure to the core of the ciphering process. In this study, a binary form of data, such as plaintext messages, or images are transformed into sequences of DNA nucleotides. Subsequently, these nucleotides pass through a Playfair encryption process based on amino-acids structure. The fundamental idea behind this encryption technique is to enforce other conventional cryptographic algorithms which proved to be broken, and also to open the door for applying the DNA and Amino Acids concepts to more conventional cryptographic algorithms to enhance their security features.

Key Words: DNA, amino acids, encryption, decryption, cryptography, security, Playfair cipher.

I. INTRODUCTION

As some of the modern cryptography algorithms (such as DES, and more recently, MD5) are broken, the new directions of information security are being sought to protect the data. The concept of using DNA computing in the fields of cryptography and steganography is a possible technology that may bring forward a new hope for powerful, or even unbreakable, algorithms.

The main purpose behind our work is to discover new fields of encoding the data in addition to the conventional used encryption algorithm in order to increase the concept of confusion and therefore increase security.

In our work, we applied the conversion of character form or binary form of data to the DNA form and then to amino acid form. Then the resulting form goes through the encryption algorithm which we chose for example; the classical Playfair cipher.

It is Adleman, with his pioneering work [5]; set the stage for the new field of bio-computing research. His main idea was to use actual chemistry to solve problems that are either unsolvable by conventional computers, or require an enormous amount of computation. By the use of DNA computing, the Data Encryption Standard (DES) cryptographic protocol can be broken [6]. In DNA steganography, A DNA encoded message is first camouflaged within the enormous complexity of human genomic DNA and then further concealed by confining this sample to a microdot[3]. Recent research considers the use of the Human genome in cryptography. In 2000, the Junior Nobel Prize was awarded to a young Romanian American student, Viviana Risca, for her work in DNA steganography.[3]

The one-time pad cryptography with DNA strands, and the research on DNA steganography (hiding messages in DNA), are shown in [2] and [3]. However, researchers in DNA cryptography are still looking at much more theory than practicality. The constraints of its high tech lab requirements and computational limitations, combined with the labor intensive extrapolation means. Thus prevent DNA computing from being of efficient use in today's security world.

Another approach is lead by Ning Kang in which he did not use real DNA computing, but just used the principle ideas in central dogma of molecular biology to develop his cryptography method. The method only simulates the transcription, splicing, and translation process of the central dogma; thus, it is a pseudo DNA cryptography method. [4]

There is another investigation conducted by [1] which is based on a conventional symmetric encryption algorithm called "Yet Another Encryption Algorithm" (YAEA) developed by Saeb and Baith [1]. In this study, he introduces the concept of using DNA computing in the fields of cryptography in order to enhance the security of cryptographic algorithms. This is considered a pioneering idea that stood behind our work in this paper.[1]

Although Playfair cipher is believed to be an old, simple and an easily breakable cipher, we believe our new modifications can make it a more powerful encryption algorithm. This is done by introducing concepts of confusion and diffusion to the core of the encryption process in addition to preserving the cipher's simplicity concept. In addition shortage in security features the plaintext message is restricted to be all upper case, without J letters, without punctuation, or even numerical values. Those problems can be easily handled in any modern cipher as handled in our new algorithm [8].

The character form of a message or any form of an image can be easily transformed to the form of bits. This binary form can be transformed to DNA form through many encoding techniques implemented in previous work and summarized in [7].

Playfair is based on the English alphabetical letters, so preserving this concept, we will use the English alphabet but from an indirect way. DNA contains four bases that can be given an abbreviation of only four letters (adenine (A), cytosine (C), guanine (G) and thymine (T)). On the other side, we have 20 amino acids with additional 3 codons to represent the Stop of coding region. Each amino acid is abbreviated by a single English character. So we are able to stretch these 20 characters to 26 characters, we will be able to represent the English alphabet.

Then, we have to convert the DNA form of data to amino acid form so that it can go through a classical Playfair cipher. Through this conversion process, we have to keep in mind the problem of ambiguity; that most amino acids are given more than possible codon. The rest of the paper is organized as follows: section two will give a brief explanation of what is DNA and process of Transcription and translation. Section three introduces our new algorithm followed by a detailed explanation of the encryption/ decryption processes. Section four shows the experiment steps, results. Section five introduces some additional security features. Section six shows conclusion and future work.

II. OVERVIEW OF DNA

A. What is Deoxyribonucleic acid 'DNA'?
DNA is a nucleic acid that contains the genetic instructions used in the development and functioning of all known living organisms and some viruses. The main role of DNA molecules is the long-term storage of information. DNA is often compared to a set of blueprints or a recipe, or a code, since it contains the instructions needed to construct other components of cells, such as proteins and RNA molecules. The DNA segments that carry this genetic information are called genes, but other DNA sequences have structural purposes, or are involved in regulating the use of this genetic information.

The DNA double helix is stabilized by hydrogen bonds between the bases attached to the two strands. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). These four bases are attached to the sugar/phosphate to form the complete nucleotide, as shown for adenosine monophosphate.

B. The genetic code
The genetic code consists of 64 triplets of nucleotides. These triplets are called codons. With three exceptions, each codon encodes for one of the 20 amino acids used in the synthesis of proteins. That produces some redundancy in the code: most of the amino acids being encoded by more than one codon.
The genetic code can be expressed as either RNA codons or DNA codons. RNA codons occur in messenger RNA (mRNA) and are the codons that are actually "read" during the synthesis of polypeptides (the process called translation). But each mRNA molecule acquires its sequence of nucleotides by transcription from the corresponding gene.

The DNA Codons is read the same as the RNA codons Except that the nucleotide thymine (T) is found in place of uridine (U). So in DNA codons we have (TCAG) and in RNA codons, we have (UCTG).

C. Transcription and translation
A gene is a sequence of DNA that contains genetic information and can influence the phenotype of an organism. Within a gene, the sequence of bases along a DNA strand defines a messenger RNA sequence, which then defines one or more protein sequences. The relationship between the nucleotide sequences of genes and the amino-acid sequences of proteins is determined by the rules of translation, known collectively as the genetic code. The genetic code consists of three-letter 'words' called codons formed from a sequence of three nucleotides (e.g. ACT, CAG, TTT).

In transcription, the codons of a gene are copied into messenger RNA by RNA polymerase. This RNA copy is then decoded by a ribosome that reads the RNA sequence by base-pairing the messenger RNA to transfer RNA, which carries amino acids. Since there are 4 bases in 3-letter combinations, there are 64 possible codons (43 combinations). These encode the twenty standard amino acids, giving most amino acids more than one possible codon. There are also three 'stop' or 'nonsense' codons signifying the end of the coding region; these are the TAA, TGA and TAG codons.

III. DNA-BASED PLAYFAIR ALGORITHM

A. Encryption algorithm of DNA-based Playfair cipher:
Playfair used to be applied to English alphabet characters of plaintext. It was unable to encode any special characters or numbers which is considered a severe drawback that enforces the sender to write everything in the English letters. This problem appears while sending numerical data, equations or symbols.

On the contrary, in our algorithm, we can use any numbers, special characters or even spaces (not preferred) in or plaintext. The encryption process starts by the binary form of data (message or image) which is transferred to DNA form according to Table 1. Then the DNA form is transferred to the Amino acids form according to Table 2 which is a standard universal table of Amino acids and their codons representation in the form of DNA [RNA codon table, Wikipedia: http://en.wikipedia.org/wiki/Genetic_code#cite_note-pmid19056476-8].

Note that each amino acid has a name, abbreviation, and a single character symbol. This character symbol is what we will use in our algorithm

Table I: DNA Representation of bits.

Bit 1	Bit 2	DNA
0	0	A
0	1	C
1	0	G
1	1	T

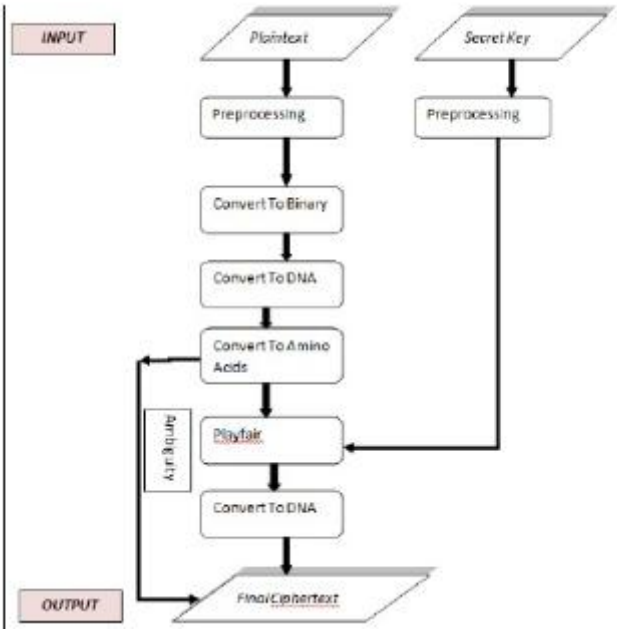

Fig. 1.flowchart of the DNA based Playfair Algorithm

B. Constructing the alphabet table:

In the table, we have only 20 amino acids in addition to 1 start and 1 stop. While we need 25 letters to construct the Playfair matrix (note that I/J are assigned to one cell).

The letters we need to fill are (B, O, U, X, Z). So we will make these characters share some amino acids their codons. The start codon is repeated with amino acid (M) so we will not use it. We will assign to (B) the 3 stop codons. We have 3 amino acids (L, R, S) having 6 codons. By noticing the sequence of DNA of each, we can figure out that each has 4 codons of the same type and 2 of another type. Those 2 of the other type are shifted to the letters (O, U, X) respectively. Letter (Z) will take one codon from (Y), so that Y: UAU, Z: UAC. Now the new distribution of codons is illustrated in Table 3.

Counting the number of codons of each character, we will find the number varies between 1 and 4 codons per character. We will call this number 'Ambiguity' of the character [AMBIG].

Now we have the distribution of the complete English alphabet, so a message in the form of Amino Acids can go through traditional Playfair cipher process using the secret key.

Table II: Amino acids and their 64 codons

Ala/A	GCU, GCC, GCA, GCG	**Leu/L**	UUA, UUG, CUU, CUC, CUA, CUG
Arg/R	CGU,CGC, CGA,CGG, AGA, AGG	**Lys/K**	AAA, AAG
Asn/N	AAU, AAC	**Met/M**	AUG
Asp/D	GAU, GAC	**Phe/F**	UUU, UUC
Cys/C	UGU, UGC	**Pro/P**	CCU,CCC, CCA, CCG
Gln/Q	CAA, CAG	**Ser/S**	UCU, UCC, UCA, UCG, AGU, AGC
Glu/E	GAA, GAG	**Thr/T**	ACU, ACC, ACA, ACG
Gly/G	GGU,GGC, GGA,GGG	**Trp/W**	UGG
His/H	CAU, CAC	**Tyr/Y**	UAU, UAC
Ile/I	AUU, AUC, AUA	**Val/V**	GUU, GUC, GUA, GUG
START	AUG	**STOP**	UAA, UGA, UAG

Table III: New distribution of the alphabet with the corresponding new codons:

	STOP										from			To			from	from		to			to	from	To
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4	3	2	2	2	2	4	2	3		2	6	1	2		4	2	6	6	4		4	1		2	
GCU	UAA	UGU	GAU	GAA	UUU	GGU	CAU	AUU		AAA	UUA	AUG	AAU	UUA	CCU	CAA	CGU	UCU	ACU	AGA	GUU	UGG	AGU	UAU	UAC
GCC	UAG	UGC	GAC	GAG	UUC	GGC	CAC	AUC		AAG	UUG		AAC	UUG	CCC	CAG	CGC	UCC	ACC	AGG	GUC		AGC	UAC	
GCA	UGA					GGA		AUA			CUU				CCA		CGA	UCA	ACA		GUA				
GCG						GGG					CUC				CCG		CGG	UCG	ACG		GUG				
											CUA						AGA	AGU							
											CUG						AGG	AGC							
4	3	2	2	2	2	4	2	3	3	2	4	1	2	2	4	2	4	4	4	2	4	1	2	1	1

Table IV: New Distribution for codons on English alphabet

A	GCU, GCC, GCA, GCG	L	CUU, CUC, CUA, CUG
R	CGU, CGC, CGA, CGG	K	AAA, AAG
N	AAU, AAC	M	AUG
D	GAU, GAC	F	UUU, UUC
C	UGU, UGC	P	CCU, CCC, CCA, CCG
Q	CAA, CAG	S	UCU, UCC, UCA, UCG
E	GAA, GAG	T	ACU, ACC, ACA, ACG
G	GGU, GGC, GGA, GGG	W	UGG
H	CAU, CAC	Y	UAU, UAC
I	AUU, AUC, AUA	V	GUU, GUC, GUA, GUG
B	UAA, UGA, UAG	O	UUA, UUG
U	AGA, AGG	X	AGU, AGC
Z	UAC		

The output form is the amino acid form of cipher text. DNA form of cipher text can be demonstrated also from Table 4 by choosing random codons accompanied to each character. The concept that one character can have more than one DNA representation is itself an addition to confusion concept that enhances the algorithm strength. Table IV shows the new distribution of codons on the amino acids and additional alphabetical English letters according to our algorithm.

C. Decryption and Ambiguity problem

The decryption process is simply the inverse of the encryption process unless that we will find a problem in constructing the DNA form of plaintext from the amino acid form which is of length (L). The problem is that we are unable to choose which codon to put in accordance to each amino acid character. This is simply the problem of codon-amino acid mapping problem arised with other algorithm based on the concept of Central Dogma like [4]. The way Nang handled this problem is to put this codon-amino acid mapping in the secret key to be sent through a secure channel [4]. This idea is not efficient since it increases the size of the key in relation to size of the plaintext.

The solution in our algorithm is located in two additional bits for each amino acid character to demonstrate which codon to choose. We said before that each amino acid has 1, 2, 3 or 4

codons to represent it. This is a number that can be put in 2 bits from 0→3.

These 2 bits can be converted to DNA form from Table 1. That is why the final cipher text is both the DNA form of cipher text of length (3L) and the array carrying the ambiguity of length (L).

In decryption, the amino acid form of plaintext with the assistance of the ambiguity array can construct the correct form of plaintext in DNA form which can be transferred to binary form and then the final character form.

D. Pseudo-code

Input:

[P] Plaintext (characters with spaces, numbers or any special characters).

[K] Secret key (English characters without any number or special characters).

Algorithm body:

Preprocessing:

1- Prepare the secret key:
   - Remove any spaces or repeated characters from [K].
   - Put the remaining characters in the UPPER case form. [K]→UPPER[K].

2- Prepare the plaintext:

- Remove the spaces from [P] (done to avoid attacker's trace to a character which is repeated many times within the message)

Processing:

1- Binary form [BP] = BINARY [P] (Replace each character by its binary representation-8 bits-)

2- DNA form [DP] = DNA [BP] (Replace each 2 bits by their DNA representation)

3- Amino acids form [AP] = AMINO [DP] (Replace each 3 DNA characters by their Amino acid character keeping in track the ambiguity of each Amino acid [AMBIG].

4- Construct the Playfair 5X5 matrix and add [K] row by row, then add the rest of alphabet characters.

5- Amino acid of cipher text [AC]= PLAYFAIR [AP].

6- DNA form of cipher text [DC] = DNA [AC].

Output:

Add [DC] and [AMBIG] together in the suitable form→final cipher text [C].

E. Samples of the program steps and output

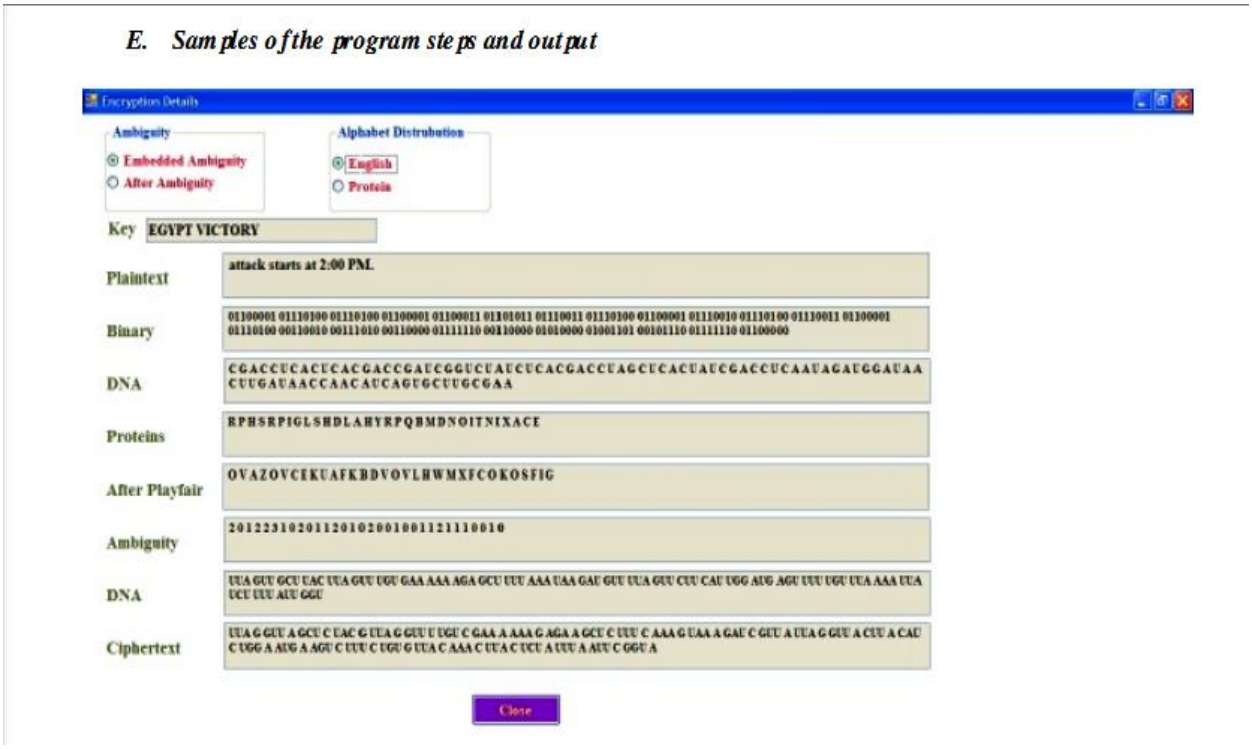

Figure 2. Sample of steps of encryption implementation

Figure 2: sample of steps of encryption implementation

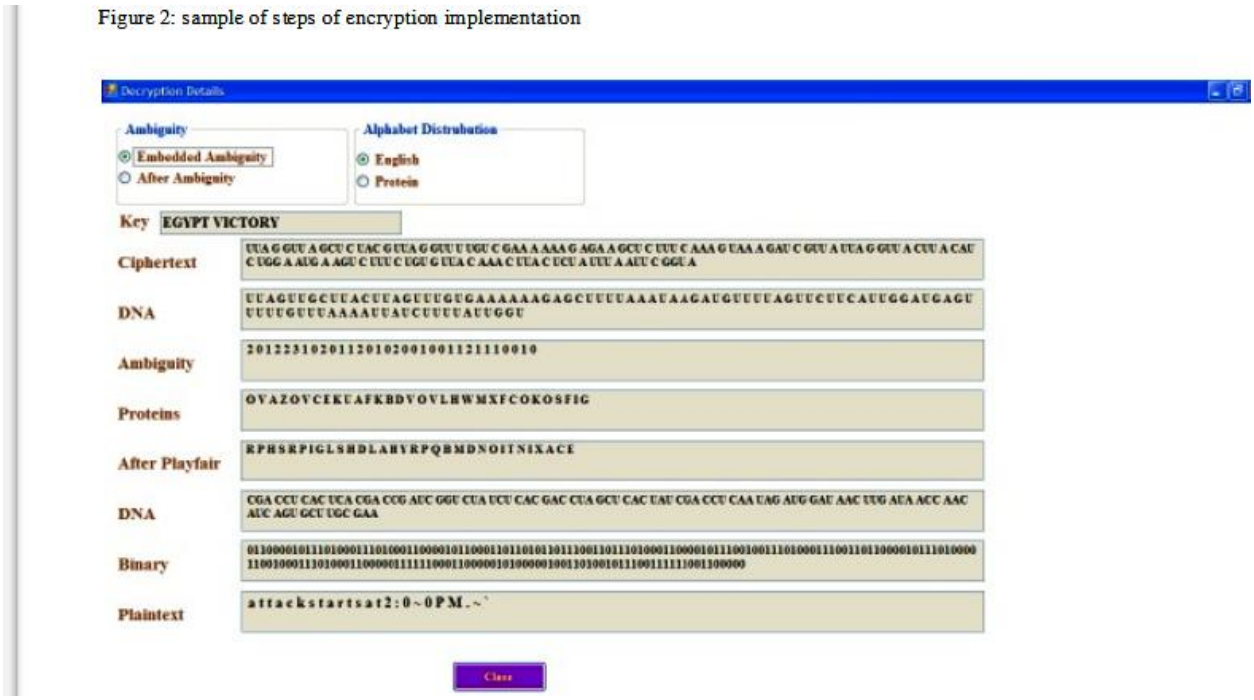

Figure 3. Sample of steps of decryption implementation

IV. EXPERIMENT AND PERFORMANCE ANALYSIS

A. Experiment

1- Experiment inputs and attributes

We will take paragraphs from the beginning of the novel according to the estimated storage size in Kilobytes (from 1 KB and increasing till 150 KB).

2-System Parameters

The experiments are conducted using Intel(R) Core (TM) 2CPU T5300, 1.73 GHz, 32 bit processor with 1GB of RAM. The simulation program is compiled using the default settings in .NET 2005 visual studio for C# windows applications under WINDOWS XP as the operating system. The experiments will be performed several times to assure that the results are consistent and valid.

3- Experiment Factors

The chosen factor here to determine the performance is the algorithm's speed to encrypt data blocks of various sizes. Suppose we will use the original sequence of English alphabet and embed the ambiguity inside the message not after it. The secret key used is "CHARLES DICKENS" which results in 11Bytes key.

4- Experiment steps:

Experiment preprocessing:

1- Loading the table of the 64 amino acids with their DNA Encodings and number of ambiguous encodings.

2- Formatting the secret key by removing spaces, repeated characters and non English letters.

3 - Formatting the plaintext by removing spaces between words and separating the repeated doubles by the character '~' which chosen to be a rarely used character.

Processing:

This includes:

1- Converting characters to binary form.

2- Converting binary to DNA

3- Converting the DNA to amino acids and recording ambiguity.

4- Do Playfair encryption.

5- Convert the amino acid form of cipher text to DNA form in addition to embedding the ambiguity in the DNA format.

6 – Experiment Results

The next table illustrates the experiments and time taken to encrypt each piece of plaintext (each is of different data loads) in milliseconds. The time taken by loading the amino acids table and preparing the secret key is ignored because it is comparatively small to processing time.

Table 4: performance results of DNA-based Playfair algorithm

Input size of plaintext (in KB)	Plaintext after preprocessing	Preprocessing plaintext	From Binary to Amino Acids form	playfair	Prepare ciphertext	Total processing time	Bytes/Second
1 (1,022 B)	846B	0	0	0	15.625	15.625	65.408
10 (9,757 B)	8124B	62.500	15.625	0	125.000	203.125	48.034
20 (20,023 B)	16599B	203.125	15.625	0	171.875	390.625	51.259
50 (50,432 B)	41781B	1062.500	46.875	15.625	437.500	1562.500	32.276
100 (97,072 B)	83910B	4687.500	78.125	31.25	859.375	5656.250	17.162
150 (153,418 B)	127098 B	11390.625	140.625	31.25	1343.750	12906.25	11.887

V. Additional security features

We have illustrated the main core of the algorithm and now we are going to suggest some additional features to the algorithm which can enhance its security and strength.

A- The key:
It is quite clear that the more random and long the key is, the more the difficulty to break the cipher will be.

B- Use Amino acids alphabet sequence instead of English alphabetical sequence:
The standard table of amino acids has a special sequence defined in the matrix [4X4] (UCAG) X (UCAG). This sequence of acids can be used instead of the sequence of English alphabet letters to fill the rest of the 5X5 matrix after adding the secret key.

C- Combine the total resulting message into long strand of DNA to be inserted in a microdot (steganography):
One of the advantages of this algorithm is the variety of ways we can use to write down the cipher text. It can be written in DNA form, binary form or even character form which is more confusing. The advantage of DNA form is that it can make us of several steganography techniques developed for DNA messages [3]. It can also be prepared in biological labs like in [2] in which DNA message goes through a biological DNA encryption process using one time pad or substitution.

D- Ambiguity a problem that contains useful confusion feature:
Some characters in table 2 can have 6 codons representing the problem of ambiguity. The way we handled the preparation of table 3 made each character have in maximum 4 codons. The number 4 can be represented by 2 bits and therefore can be represented by one DNA character. That was a benefit that made us able to write the cipher text with ambiguity in the form of DNA.

E- Use of conventional XOR-ing procedure:
Another way to increase security is defining another key that can be XOR-ed with the amino acid form or DNA form of cipher text. It was a pioneer idea by [1] to choose the key as the DNA strand of a certain organism. This idea assures the key randomness and variety in length according to the length of the message.

VI. CONCLUSION AND FUTURE WORK

The fundamental idea behind this technique is to open the door for the idea of applying the DNA and Amino Acids encoding concepts to other conventional cryptographic algorithms to enhance their security vulnerability- features. Our algorithm initially succeeded in overcoming some main problems in "Playfair cipher" like restriction of plaintext to "English Alphabet". As in our algorithm the plaintext is to be converted to its binary value before encryption, it now clear that the plaintext message can be written in upper or lower case, with any punctuation, and numerical values.
Other papers conducted the idea of amino acids way of representation from the point of view of the central dogma design [4]. But they were unable to clearly handle the problem of ambiguity as performed by our algorithm. Our algorithm made few preprocessing steps to handle this problem and the result was quite accurate (same input message obtained after decryption). This feature is very important when regarding an encryption algorithm in order to verify the concept of data integrity or in other words, to assure that data after decryption to be the same input data before encryption. Finally, our algorithm provides different forms of the cipher text like: Binary form, DNA form, Amino Acid form or character form. Those various forms can match different used applications.
Our future work is dedicated to implementing this encoding on other known algorithms and measuring its performance and security, Also, Experiments should be conducted to implement the algorithm on different applications to ensure its feasibility and applicability.

REFERENCES

[1] Sherif T. Amin, Magdy Saeb, Salah El-Gindi, "A DNA-based Implementation of YAEA Encryption Algorithm,"I AST ED International Conference on Computational Intelligence (CI 2006), San Francisco, Nov. 20, 2006.

[2] Ashish Gehani, Thomas LaBean and John Reif. DNA-Based Cryptography. DIMACS DNA Based Computers V, American Mathematical Society, 2000.

[3] TAYLOR Clelland Catherine, Viviana Risca, Carter Bancroft, 1999, "Hiding Messages in DNA Microdots". Nature Magazine Vol.. 399, June 10, 1999.

[4] KANG Ning, "A Pseudo DNA Cryptography Method", Independent Research Study Project for CS5231, October 2004.

[5] Leonard Adleman. "Molecular Computation of Solutions to Combinatorial Problems". Science, 266:1021-1024, November 1994.

[6] Dan Boneh, Cristopher Dunworth, and Richard Lipton. "Breaking DES Using a Molecular Computer". Technical Report CS-TR-489-95, Department of Computer Science, Princeton University, USA, 1995.

[7] Dominik Heider and Angelika Barnekow, "DNA-based watermarks using the DNA-Crypt algorithm", Published: 29 May 2007 BMC Bioinformatics 2007,8:176 doi:10.1186/1471-2105-8-176, Barnekow; licensee BioMed Central Ltd.

[8] William Stallings."Cryptography and Network Security", Third Edition, Prentice Hall International, 2003

Research Paper by-

Mona Sabry, Mohamed Hashem, Taymoor Nazmi, Mohamed Essam Khalifa Faculty of Computer Science and Information Systems, Ain Shams University, Cairo, Egypt **Journal Published in- International Journal of Computer Science and Information Security, Vol. 8, No. 3, 2010.**