Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BINARY FUSION PROCESS TO THE CIPHERING SYSTEM "SEC EXTENSION TO BINARY BLOCKS"

[1]ZAKARIA KADDOURI, [2]FOUZIA OMARY AND [3]ABDOLLAH ABOUCHOUAR
[1] PhD student, Department of Computer Sciences. Faculty of Sciences, Mohamed V University, Rabat.
[2] Professor, Department of Computer Sciences. Faculty of Sciences, Mohamed V University, Rabat.
[3] PhD student, Department of Computer Sciences. Faculty of Sciences, Mohamed V University, Rabat.

LRI Laboratory (Ex: Networks and Data Mining Laboratory)
Department of Computer Science, Mohamed V University – Faculty of Sciences Rabat – Morocco.
E-mail : [1]kaddouri.zakaria@gmail.com, [2]omary@fsr.ac.ma, [3]abdollah.abouchouar@gmail.com

ABSTRACT

In this paper, we present a new symmetrical encryption system based on the binary extension of the symmetric encryption system SEC, which transforms the encryption issue into a combinatorial optimization problem using basic tools such as evolutionary algorithms. Our main objective is to change the appearance frequencies of plaintext binary blocks, to make statistical cryptanalysis impossible. We designed a new encryption method called "Binary fusion Process (BFP)", it is a preparatory step for the application of evolutionary algorithm, and can in fact generate a more interesting initial population. Through the key generated by our algorithm, we illustrate the process of encryption and decryption, and then we present our various applications while interpreting them. Finally, to evaluate our system we compare it to other well known systems. Moreover, the experimented results show that the robustness of our system is undeniable to any attack by frequency analysis.

Keywords: *Symmetric Encryption, Evolutionary Algorithms, Combinatorial Optimization, SEC Extension To Binary Blocks, Frequency Analysis.*

1. INTRODUCTION

Cryptography is the process of transcribing intelligible information to an unintelligible one by the application of secret conventions whose effect is reversible, i.e. transform with an encryption key a clear text into a ciphertext, so that the reverse transformation is only possible with the knowledge of the decryption key. Symmetric cryptography is the oldest form of encryption, it includes algorithms for which sender and receiver share the same secret key.

Evolutionary algorithms are a family of algorithms inspired by the theory of evolution to solve various problems. Thus, they evolve a set of solutions to a given problem, with a view to find the best results. These are stochastic algorithms, as they use random process iteratively. We note that they are useful in cryptography. The effectiveness of these algorithms is proved by solving combinatorial optimization problems, which are generally NP-complete or NP-hard [9].

The symmetric encryption system "SEC extension to binary blocks" is based on evolutionary algorithm, whose main objective is to establish an exchange of the appearance frequencies of the different binary blocks belonging to the message to be encrypted, as well as their own positions.

The main objective of this work is to strengthen the resistance of the encryption system "SEC extension to binary blocks". The Binary Fusion process is introduced to change the appearance frequencies of binary blocks, and to reach equilibrium.

This paper is organized as follows. The next section describes evolutionary algorithms, the second part defines the binary fusion process. Experimental results and evaluative discussion are given respectively in sections 3,4.

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. EVOLUTIONARY ALGORITHMS

2.1 Definition

Basically, the metaheuristics consist of creating the evolution of a baseline configuration by replacing it repeatedly by a new configuration chosen in its neighborhood. The evolutionary algorithm is a kind of the metaheuristics. It is useful for optimization when other techniques such as gradient descent or direct, analytical discovery are not possible. It incorporates aspects of natural selection or survival of the fittest and maintains a population of structures (usually randomly generated initially), that evolves according to rules of selection, recombination, mutation and survival, referred to as genetic operators. A shared "environment" determines the fitness or performance of each individual in the population. The fittest individuals are more likely to be selected for reproduction (retention or duplication), while recombination and mutation modify those individuals, yielding potentially superior ones [5].

We present the mechanisms involved in evolutionary algorithms in the flowchart in Figure1.

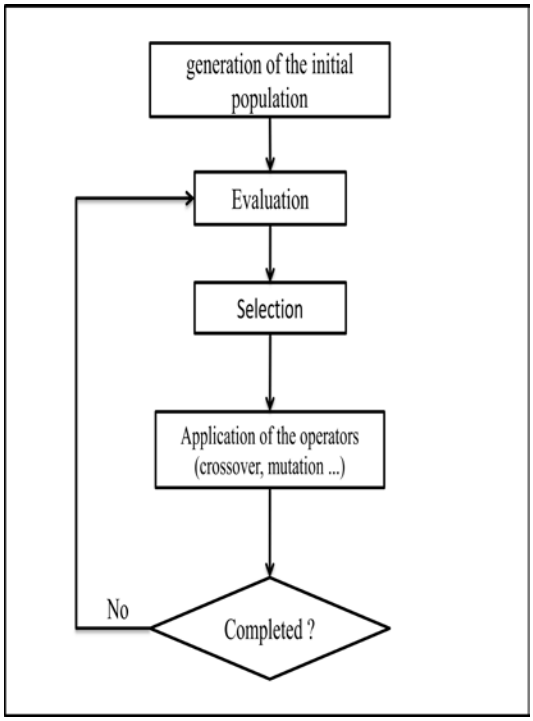

Figure 1: Flowchart of an evolutionary algorithm

In precisely, the contribution of our work consists of the design and implementation of symmetric encryption systems with AE as basic tools.

2.2 Algorithm

An evolutionary algorithm generates an initial population P of μ individuals, and then makes the population P (generations) evolves following a repeated pattern.

• For every generation:

1) Select for reproduction: is chosen in the current population P, λ individuals which become parents (mating pool P').

2) Vary operators: by applying the crossover and mutation on the individuals of P, a population P" of λ children is obtained.

3) Evaluate the performance of elements of P" (children).

4) Select for survival: the elements of P (current population) and P" (children) the μ individuals are chosen among these elements which will generate the population P of the next generation.

3. DESCRIPTION OF "SEC EXTENSION TO BINARY BLOCKS"

In order to avoid methods of classical cryptography, target of cryptanalysis attacks, an encryption technique was introduced to modify the text characters and then their appearance frequencies starting their coding in binary [1].

3.1 Description of the method

Start with a binary coding of the text characters to be encrypted and prescribe an integer k (k> 1).

We can then consider the text as a series of blocks of k bits, called k-blocks. And we establish the respective lists of occurrences (or positions) of different k-blocks in the text. Then, we apply the AE used by the SEC lists obtained to reach a number in binary text. And finally, we convert the binary ciphertext characters according to the encoding used.

Let T be the message to be encrypted. T is a sequence of n blocks.

We notice that: $L_i \cap L_j = \emptyset$ if $i \neq j, \forall i, j \in \{1, 2, \ldots, m\}$.

Then $L_1, L_2, \ldots, L_m$ is a partition of the set $\{1, 2, \ldots, n\}$.

The message T can be represented by the vector below:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(B_1,L_1)	(B_2,L_2)	...	(B_n,L_n)

The goal of "SEC extension to binary blocks" is to establish the maximum disorders over the positions of binary blocks. In order to make it happen, we must repeatedly change the distribution lists over the different blocks of T. In other words, we must find a permutation σ of {1, 2, …, m} such that the difference between the cardinal of the new list $L_{\sigma(i)}$ assigned to the block B_i and the cardinal of the initial list L_i reaches the maximum level. In this case, we are confronted with a combinatorial optimization problem. Nevertheless, the evolutionary algorithms are very effective in this kind of problem. So, they will be applied to permutations problems[8].

The main objective of this work is to change at the maximum level the positions and the appearance frequencies of the binary blocks. Note that "SEC extension to binary blocks" starts with randomly generated potential solutions using the processes of classical cryptography.

3.2 Algorithm of "SEC extension to binary blocks"

3.2.1 Coding

Use an individual (or chromosome) as a vector of size m.

Genes are the lists $L_{pi}(1 \le i \le m)$.

L_{pi} is the i^{th} gene which contains the new positions that will take the block B_i.

3.2.2 Initialization

Creation of the initial population P_0 consists of q individuals: $X_1, X_2, \ldots, X_q$.

Let Original-Ch be the chromosome which genes are $L_1, L_2, \ldots,$ and L_m lists (placed in this order).These lists represent the message's binary encoding before the application of the algorithm. We apply q permutations on Original-Ch in order to get an initial population formed by q potential solutions.

Set i :=0;

3.2.3 Evaluation of individuals

Let X_j is an individual of P_i whose genes are: $L_{j1},L_{j2},\ldots,L_{jm}$.

The evaluation function F is defined on the set of individuals X_j by:

$$F(X_j) = \sum_{i=1}^{m} | \text{card}(Lji) - \text{card}(Li) |$$

3.2.4 Selection of the best individuals

The conventional method of the roulette wheel retains the strongest individuals. A Control function is introduced to eliminate the individuals in whom the values of only a minority of genes have changed in comparison with the initial chromosome: Original-Ch.

Since this problem is narrowed to a permutation problem with constraints, the genetic operators have been adjusted to this kind of problems [6].

3.2.5 Crossover MPX (Maximal Preservative X)

This cross is applied to selected individuals with a very precise rate. The best rate is from 60% to 100%.

3.2.6 Transposition Mutation

Choosing the mutation consists of randomly swapping two genes of a chromosome. This operator is applied to individuals from crossing with an appropriate rate, preferably from 0.1% to 5%.

Place new offspring in a new population P_{i+1}.

Repeat steps 2, 3 and 4 until a stopping condition.

3.2.7 Stopping condition

The function F is bounded because $0 \le F(X) \le 2*m$, for each individual X. In fact, the function F admits a maximum since it is bounded. According to some researches, the convergence result of fitness function is made but it can be a value close to Max, which can be experimentally determined. Final-Ch denotes the final solution given by our evolutionary algorithm.

Final-Ch denotes the final solution given by our evolutionary algorithm. From Original-Ch and Final-Ch, the symmetric key is constructed. This key is called a genetic key [8].

4. DEFINITION OF THE NEW SYSTEM USING THE BINARY FUSION PROCESS (BFP)

The Binary Fusion Process (BFP) consists of creating equilibrium between the lists of positions of binary blocks that represent the genes of the system. The resistance of the system against attack by frequency analysis will be stronger and the system becomes more robust.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We apply this process on the binary blocks of the plaintext, which will generate an additional secret key which we call the fusion binary key, then we apply the "SEC extension to binary blocks." Figure 2. shows a presentation of our system.

4.1 Description

Let M be a plaintext that we code into binary and cut out in blocks B_1, B_2,…B_n of the same size k (if the last block contains less than k bits, we complete it by bits of « 0 »). Then, we determine the positions of the various blocks constituting this plaintext, and instead of applying the evolutionary ciphering algorithm to the lists of binary blocks positions in the plaintext like in [2], we first apply the binary fusion process to the various blocks of the binary coded text.

• Fusion lists

The message "T" obtained in the first step is composed of blocks bit B_1, ..., B_n, which are associated with lists L_1, L_2, ..., L_n. These lists usually have different sizes. The objective of the first part of our system is to merge the lists. In other words, we merge the lists of small and medium sized lists in others with large sizes. This will change the maximum appearance frequencies of binary blocks in the text and will establish more disorder in their positions.

• Encryption

The second part of the encryption system is to apply the "SEC extension to binary blocks" that we saw earlier. Thus, we obtain an encrypted message T '.

4.2 Formalization of the problem

4.2.1 Ciphering

First part: the fusion

We sort the set of the lists L_1, L_2, …, L_n according to their sizes in the decreasing order then we divide it in tree subsets of sizes near to [n/3] (floor of n/3), named respectively : E_L, E_m, E_p.

Let us indicate respectively by N_L, N_m and N_p the cardinals of E_L, E_m, E_p. The process fusion is recursive. It is described below.

Let us indicate by L_m and L_p the smallest lists of E_m, E_p respectively and by S_k the desirable key size

-If the merge of these tow lists brings to a key of undesirable size then fusion is applied only in E_p as follows:

- Let us take randomly a number of lists L_{p1}, L_{p2},…, L_{pf} in E_p.
- applying fusion to these lists means:

-Replace the blocs B_{p1}, B_{p2},…,B_{pf} corresponding to these lists by one bloc B_f chosen and representing any other list.

-Add the triplet:

$([B_{p1}, B_{p2},\ldots, B_{pf}] ; [L_{p1}, L_{p2}, \ldots, L_{pf}] ; B_f)$

to the fusion key.

$E_p \leftarrow E_p - \{ L_{p1}, L_{p2},\ldots, L_{pf} \}$.

-Repeat the application of fusion on E_p until reaching the size of the desired key.

-Else the fusion will be applied to E_m U E_p as follows:

- Let us indicate initially the lists which we indeed to merge. These lists are composed of L_{p1}, L_{m1},…, L_{pf}, L_{mf} such as f is taken randomly in $\{1,2,\ldots, \min(N_m,N_p)\}$, L_{p1}, L_{p2},…, L_{pf} are selected in E_p in increasing order of their size, in alternation with L_{m1}, L_{m2},…, L_{mf} which are taken in E_m in the same order , while taking account of the following iterative processing:

- r ←1; E←∅
- repeat

$E \leftarrow E \cup L_{pr} \cup L_{mr}$

If size of E ≤ S_k then r ←r+1

Until r=f or size of E is bigger than S_k

- f ← r

-Applying fusion to the lists above means:

- Replace the Blocs B_{p1}, B_{m1}, B_{p2}, B_{m2}, …, B_{pf} , B_{mf} corresponding to these lists by one bloc B_f chosen randomly and not representing any list.
- Add the triplet:

$([B_{p1}, B_{m1}, B_{p2}, B_{m2}, \ldots, B_{pf}, B_{mf}] ; [L_{p1}, L_{m1},\ldots, L_{pf}, L_{mf}] ; B_f)$ To the fusion key.

-Repeat the process of fusion on:

$E_m \leftarrow E_m - \{ L_{m1},\ldots, L_{mf} \}$; $E_p \leftarrow E_p - \{ L_p,\ldots, L_{pf} \}$

Until reaching the size of the desired key S_k.

- If we indicate by F_O the number of fusion applied then the generated key representative of these operations is a set of F_O triplet of the from:

$([B^i_1, B^i_2, \ldots B^i_{pi}], [L^i_1, L^i_2, \ldots L^i_{pi}], B_i)$

Where Bi is the substitute of the blocs B^i_1, B^i_2,…,B^i_{pi} whose lists of respective positions in the plaintext are : L_{i1}, L_{i2}, …, L_{ipi}.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Thus after fusion, the number of blocs (thus also of lists) is reduced to m (m<n). And the new ciphered text, T_f, will be denoted by the following:

(B'_{f1},L_{f1})	(B'_{f2},L_{f2})	...	(B'_{fm},L_{fm})

Second Part: Application of Our Evolutionary Algorithm (see chapter 3.2)

4.2.2 Deciphering

We start by deciphering the second part of the system. We represent the encoded text T' by a vector of list. Let's by B'_1, B'_2, …, B'_m the different blocs of T' and by L'_1,L'_2,…, L'_m their respective lists of positions. Thanks to the genetic key the blocks are going to recover their lists of corresponding positions in the text T_f obtained after the first part of ciphering [8].

Indeed, the key can be represented by a vector, that we denote Key, of size m such that:

key(1)=p1,key(2)=p2,…, key(i)=1, …, key (m)=pm

where:

The bloc B'_{p1} is going to be associated to the list L'_1.

The bloc B'_{p2} is going to be associated to the list L'_2.

…

The bloc B'_{m1} is going to be associated to the list L'_m.

Thus we get the text T_f.

Then, thanks to the fusion key which is clear and direct, we can immediately re-after the merged lists L_{fi} into sub-lists of origin and assign to each of latter its corresponding bloc. Thus we obtain initial text T.

5. EXPERIMENTATIONS

5.1 Configuration

We apply our system on texts of different sizes and for each of them; we attempt to find the best parameters that can give an optimal solution. Then we record the important results to be known: average of convergence of the fitness function and number of generations reached at the time of this convergence.

These results are mentioned in Table 1.

Table 1: Summary Statement Of Results

Size of Plaintext	Size of blocks / Population		5	6	7	9	10	11	12
1 000 characters	20	Number of generations	65	32	35	38	36	50	87
		Convergence	0	0	0	0	0	99	0
	30	Number of generations	71	62	68	79	76	91	101
		Convergence	27	44	20	0	0	0	4
	40	Number of generations	87	77	69	80	92	99	109
		Convergence	33	54	99	4	0	0	0
3 000 characters	20	Number of generations	51	78	68	103	76	110	121
		Convergence	22	0	0	92	0	55	0
	30	Number of generations	49	73	86	97	144	132	140
		Convergence	0	0	76	55	0	94	0
	40	Number of generations	60	49	56	94	88	130	132
		Convergence	0	98	22	14	0	0	88
6 000 characters	20	Number of generations	50	64	84	95	86	93	97
		Convergence	0	0	29	0	0	0	72
	30	Number of generations	77	86	64	58	92	98	106
		Convergence	0	34	36	0	0	59	1
	40	Number of generations	96	90	73	87	122	104	140
		Convergence	33	0	0	32	90	7	21
10 000 characters	20	Number of generations	44	53	66	83	87	89	101
		Convergence	5	0	39	75	0	53	36
	30	Number of generations	56	54	63	84	59	88	109
		Convergence	44	0	0	0	33	0	34
	40	Number of generations	88	95	108	97	100	130	150
		Convergence	0	22	0	54	0	0	88

We have noticed that the best results were found such as:

- Size of blocks is: 6
- Population size is: 20.

5.2 Key lengths calculation

For symmetric keys, encryption strength is often described in terms of the size or length of the used keys: key length is measured in bits and longer keys generally provide stronger encryption. The key of our system is composed of three elements: genetic key, the binary fusion key and the size of blocks "k".

- Denote by M the average of the various blocks used in the fusion binary and by L the average size of the merged lists and by k the block size and we indicate by Fo the number of fusion applied.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Denote by Tr the average size of a triplet and K_f length of the binary fusion key :
$Tr = ((M*k)+(M*8*L)+k)$.

- Hence The binary fusion key length is at most: $K_f = Tr*F_0 \leq S_k$.
Or S_K is already fixed departing and
$F_0 = [S_k / Tr]$

- The size of genetic key is the product of the number of different blocs and 8 bits.

To determine the size of keys, the table below shows a study on several texts of different sizes which help to extract the number of different blocks before and after BFP.

Table 2: Comparison Of The Number Of Binary Blocks Generated Before And After BFP

	Nbr_Binary Blocks / Text caracters	Before BFP	After BFP
K=5	1300	28	20
	3000	27	23
	6000	29	25
	10000	31	26
K=6	1300	54	40
	3000	57	41
	6000	60	43
	10000	63	45
K=7	1300	90	78
	3000	97	85
	6000	101	90
	10000	122	105
K=9	1300	188	169
	3000	198	179
	6000	224	202
	10000	250	225

From the results of k=6, we concluded that:

- The genetic key length is: 42* 8 bits=336 bits with 42 is the average of different blocks.
- The binary fusion key length is at most:

- Calculation of the average size of a triplet:

$T_r = ((16*6)+(16*8*5)+6)) = 640$

- If we take $S_k = 1500$ then $F_o = 2$.

The binary fusion key length is: $K_f = 640*2 = 1280$ bits.

Then the key length of our system is the addition of the both keys below.

If we compare these results to length key of TripleDES; which is considered as one of the best bloc cipher system; we can deduce that our system is able to resist against brute force attack more than Triple DES for a long time. Another advantage of our system compared to Triple DES is that its key is considered as session key and randomly generated by our system.

5.3 Execution time

The Second comparison is about the execution time between our new ciphering system with BFP, older ciphering system without BFP and TripleDES. The Table below gives an example of this comparison.

Table 3: Comparison to TripleDES

	System without fusion	System with fusion	Triple DES
Time of encryption	65	80	96
Time of decryption	17	18	90

Adding the BFP to our ciphering system hasn't really influenced the execution time and is still faster than TripleDES.

5.4 Comparison of the analysis frequencies

Comparing appearance frequencies is the main indicator of the performance of the new system. The table and figure below compare the appearance frequencies of the different blocks in the binary plaintext, then in the ciphertext using the BFP process and the ciphertext without BFP.

Table 3: Frequency Analysis In The Plaintext, The Ciphertext Using The BFP And The Ciphertext Without BFP.

Frequency analysis in the plain text	Frequency analysis in the text Encoding without BFP	Frequency analysis in the text Encoding with BFP
74	74	74
65	65	65
62	62	62
62	62	62
60	60	60
59	59	59
56	56	56
55	55	55

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

54	54	54
52	52	52
49	49	50
47	47	50
45	45	49
40	40	43
40	40	43
40	40	40
39	39	38
34	34	37
34	34	33
30	30	30
29	29	30
28	28	30
27	27	29
26	26	29
26	26	28
26	26	28
24	24	28
24	24	28
21	21	27
21	21	27
20	20	26
20	20	26
18	18	25
18	18	25
18	18	24
16	16	21
16	16	21
16	16	20
15	15	20
14	14	-
14	14	-
14	14	-
13	13	-
13	13	-
12	12	-
12	12	-
8	8	-
7	7	-
6	6	-
6	6	-
3	3	-
2	2	-
1	1	-
1	1	-

Actually, due to the Binary Fusion Process, the frequencies of the binary blocks are not recognized anymore; therefore, the cryptanalysis based on the study of appearance frequencies cannot rely on wrong statistics. Figure 4. shows a graphical representation of the apparition frequencies in the plaintext, the ciphertext using the BFP and the ciphertext without BFP.

As seen on the graphic above, the majority of binary blocks of the ciphertext using our new system ("SEC extension to binary blocks" with BFP) have almost the same appearance frequencies.

6. **Conclusion and perspectives**

Two goals have been achieved in this article. The first is the design of a new encryption method called "Binary Fusion Process" whose main advantage is to strengthen the system against the most threatening attacks (attack by frequency analysis and the brute force attack).

The second is the exploitation of the EA in order to conceive and realize a ciphering that benefits from all its qualities (simple genetic operations, performance,...). In addition the new system uses a variable-length encoding to represent a symbol of the data input, which allows encryption of any kind of information (text, image, sound ...).Finally, our system generates another secret key that we call "key binary fusion" which reinforces the genetic key.

Research works will be carried out by our team to find appropriate techniques that compress the sizes of the two generated keys and cipher them.

REFERENCES

[1] A.Mouloudi, F.Omary, A.Tragha, A.Bellaachia « An Extension of evolutionary Ciphering

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

System». 2006 International Conference on Hibrid Information Tchnology,Novembre 9th – 11th,2006.

[2] F.Omary, A.Tragha, A.Lbekkouri, A.Bellaachia, A.Mouloudi « An Evolutionist Algorithm to Cryptography». Brill Academic Publishers – Lecture SeriesAnd Computational Sciences Volume 4, 2005, pp.1749-1752

[3] F.Omary, A.Tragha, A.Lbekkouri, A.Bellaachia, A.Mouloudi « A New Ciphering Method Associated with Evolutionary Algorithm». Lecture Notes in Computer Science – Publisher : Springer Berlin / Heidelberg –ISSN: 0302-9743 –Subject :Computer Science-Volume 3984/ 2006.

[4] F.omary, A.Tragha, A.Bellaachia, A.Mouloudi. « Design and Evaluation of Two Symmetrical Evolutionist-Based Ciphering Algorithms ». International Journal of Computer Science and Network Security (IJCSNS) February 28, 2007 pp 181-190.

[5] Hans DelfsetHelmut Knebl, « Introduction to Cryptography :Principles and Applications».

[6] Gareth Jones , « Genetic and Evolutionary Algorithms ». University of Sheffield, UK

[7] Menezes A.J., Oorschot, P.C. van et Vanstone S.A., « Handbook of Applied Cryptography».(CRC Press, 1997).

[8] Thesis F.Omary, « Applications des algorithmes évolutionnistes à la cryptographie». University of science- Rabat 2006.

[9] Goldberg D.E, Genetic Algorithms in Search, Optimisation& Machine Learning. Addison-Wesley Publishing Company,Inc,1989.

[10] Florin G. et Natkin S.les techniques de la cryptographie.CNAM 2002.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

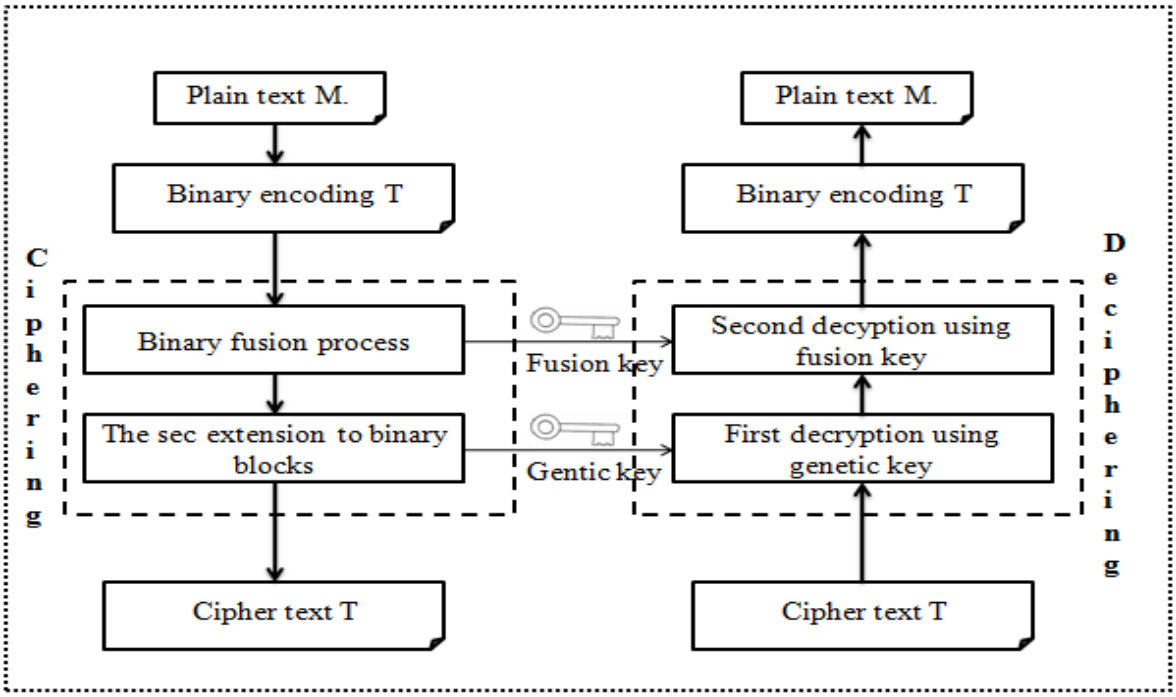

Figure 2: Schematic Of Our Encryption System

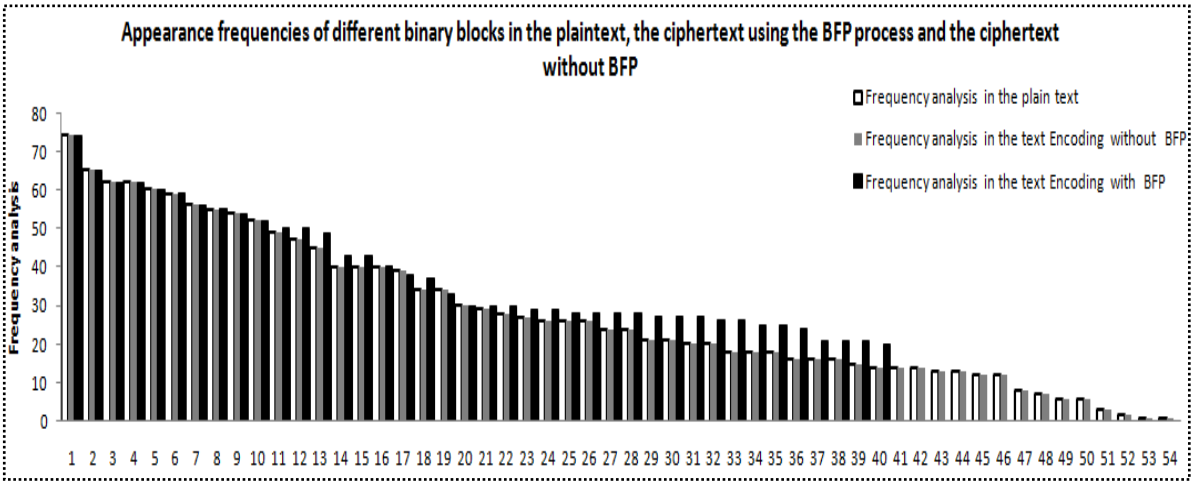

Figure 4: Graphical Representation Of The Appearance Frequencies Of Different Binary Blocks In The Plaintext, The Ciphertext Using The BFP Process And The Ciphertext Without BFP.