UACEE International Journal of Advancements in Electronics and Electrical Engineering – IJAEEE
Volume 2 : Issue 2 [ISSN 2319 – 7498]
Publication Date : 05 June 2013

New Cryptosystem Based on IDEA with Optimal Diffusion 8x8 MDS Matrix

H. Elkamchouchi, , Senior member IEEE [1]
[1]Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt
Helkamchouchi@ieee.org

M.R.M. Rizk, Senior member IEEE [2]
[2]Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt
mrmrizk@ieee.org

Fatma Ahmed, Member UACEE [3]
[3]Electrical Engineering Department, Alexandria Higher Institute of Engineering and Technology, Alexandria, Egypt.
moonyally@yahoo.com

Abstract— **The increasing ubiquity of information technologies in all aspects of human life makes security issues one of the most critical aspects of system design. In this paper we introduce a new symmetric cryptosystem based on IDEA system. The plaintext block is divided into basic sub-blocks each of thirty-two bits in length. The new Proposal can encrypt blocks of plaintext of length 512 bits into blocks of the same length. The key length is 1024 bits. The total number of rounds is 16. It uses modulo 2^{32} addition and thirty-two bits XORING are used followed by modulo $2^{32}-5$ multiplication. It uses new efficient MDS (Maximum Distance Separable) matrix which provides optimal diffusion mapping. In this system, we try to have maximum branch number for our new MDS matrix, we also try to get the minimum correlation between plaintext and ciphertext, highly avalanche effect and defeat the frequency analysis and most well-known attacks. The new algorithm is compared with IDEA and gives excellent results from the viewpoint of the security characteristics and the statistics of the ciphertext. Also, we apply the randomness test to the proposed algorithm and the results shown that the new design passes all tests which proven its security.**

***Keywords*— IDEA cryptosystem, MDS matrix, Branch number, frequency analysis.**

I. Introduction

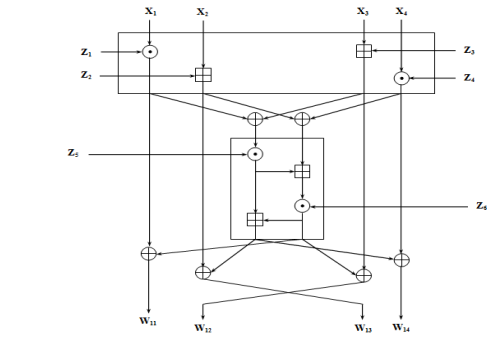
A. *IDEA*

The International Data Encryption Algorithm (IDEA) is a symmetric-key, block cipher. It was published in 1991 by Lai, Massey, and Murphy [1]. IDEA encrypts a 64-bit block of plaintext to 64-bit block of ciphertext. It uses a 128-bit key. The algorithm consists of eight identical rounds and a "half" round final transformation. The mechanism is outlined as follows:

• ***The key generation algorithm*** selects a "truly random" bit string of length 128.

• ***The encryption algorithm*** IDEA (k, x) takes a key k and a plaintext x as input. Besides some pre- and post processing that in particular splits a block into 4 quarter blocks of 16 bits each and finally recombines them, respectively, the algorithm basically performs 8 uniform rounds. Each round starts by applying a first layer of two 16-bit additions and two 16-bit multiplications on the quarter blocks and appropriate parts of the round key. Afterwards a self-inverse structure combined from two keyed 16-bit additions, two keyed 16-bits multiplications, and six 16-bit XOR operations is performed.

• ***The decryption algorithm*** is basically the same as the encryption algorithm, except that the round keys are used in the reversed order and the parts for the rounds starts and the post processing are algebraically inverted. The overall structure of IDEA is shown in Fig 1.



(a) Single Round

(b) The output transformation

Figure. 1 The overall structure of IDEA

B. *The MDS Matrix*

Maximum distance separable matrixes (MDS) are widely used in design of block ciphers and hash functions etc. Based on the character of its differential branch number, MDS matrix is widely used and the arithmetic using MDS matrixes can effective against differential cryptanalysis and linear cryptanalysis. A linear code over Galois field $GF(2^p)$ is denoted as an (n,k,d) code, where n is the symbol length of the encoded message, k is the symbol length of the original message, and d is the minimal symbol distance between any two encoded messages[2].

Definition 1: Let κ be a finite field and p and q be two integers. Let $x \mapsto M \times x$ be a mapping from K^p to K^q defined by the $q \times p$ matrix M. We say that it is a linear multipermutation (or an MDS matrix) if the set of all pairs $(x, M \times x)$ is an MDS code, i.e. a linear code of dimension p, length $p+q$ and minimal distance $q+1$ [3].

The following theorem [4] will depict the character of MDS matrix from the angle of a subdeterminant.

Theorem 1: A matrix is an MDS matrix if and only if every sub-matrix is non-singular.

MDS matrices are constructed by two types of matrices: circulant and Hadamard matrices.

Circulant matrices: Give k elements $\alpha_0, \alpha_1, ..., \alpha_{k-1}$, a circulant matrix M is constructed with each entry $M_{i,j} = \alpha_{(i+j) \bmod k}$.

Hadamard matrices: Give k elements $\alpha_0, \alpha_1, .., \alpha_{k-1}$, Hadamard matrix M is constructed with each entry $M_{i,j} = \alpha_{(i \oplus j)}$.

Definition 2: Let K^* be a set including a distinguished one denoted 1. Let M be a $q \times p$ matrix whose entries lie in K^*.

1. We let $v_1(M)$ denote the number of (i, j) pairs such that $M_{i,j}$ is equal to 1. We call it the number of occurrences of 1.

2. We let $c(M)$ be the cardinality of $\{M_{i,j}; i = 1, ..., q; j = 1, ..., p\}$. We call it the number of entries.

The following lemmas provide optimal constructions for small p and q.

Lemma 1: We have $v_1^{q,p} \geq p + 2q - 3$ for any p, q such that $q \leq p$.

Lemma 2: For any m we have $c^{2m-1, 2m-1} \leq m$.

C. *Branch Numbers of Matrices*

The branch number of a permutation function is representing the diffusion rate and measures security against differential and linear cryptanalysis [5]. The branch number is defined as the minimum number of nonzero elements in the input and output when the input elements are not all zero. The branch number of an $n \times n$ matrix M is defined by:

$$\beta(M) = \min\left\{wt(x) + wt(M.x^T) \mid x \in \left(\{0,1\}^m\right)^n, x \neq 0\right\} \quad (1)$$

Where: wt : Hamming weight. $x = (x_1, x_2, ..., x_n)^T, x_i \in \{0,1\}^m$, .

Theorem 2: For MDS $(2n, n, d)$ code over $GF(2^8)$, then the branch number of M^T is d.

We see that the maximum branch number of $n \times n$ binary matrices is equal to the maximum distance of binary linear $[n, 2n]$ codes. It is an important topic in the coding theory to find the maximum distance of binary linear $[n, 2n]$ codes.

II. The New Proposed System

The new system is a block cipher; it can encrypt blocks of plaintext of length 512 bits into blocks of the same length. The key length is 1024 bits. We test the new algorithm for many numbers of round, we found that the efficient number of round which gives better avalanche effect is 16. In the new system we: propose new efficient MDS matrix, introduce new XORed step to make the data depend on subkey of the round and to resist the frequency analysis attack and we generate the round subkey using the new MDS matrix.

A. *The New Efficient MDS Matrix*

In the new algorithm, we design new MDS matrix which provides the maximum branch number and the optimal construction conditions. The new matrix is self inverse so that same matrix can be used for decryption algorithm, which decreases the complexity of system. The new MDS matrix is 8×8 Hadamard matrix. MDS is $(16, 8, 9)$. MDS property of the matrix is calculated i.e. a $(16, 8, 9)$ code is represented MDS if d = n – k + 1.This can be done by checking the branch number of the transformation. The input with one or two active byte column is multiplied with the matrix and the output column is checked, if the total number of active bytes including input and output bytes is equal to 9 then it satisfies the property of MDS. The new MDS matrix is checked for the involution property. We design it by provide the involution conditions which can calculate from the next matrix:

$$\begin{pmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ b_1 & b_0 & b_3 & b_2 & b_5 & b_4 & b_7 & b_6 \\ b_2 & b_3 & b_0 & b_1 & b_6 & b_7 & b_4 & b_5 \\ b_3 & b_2 & b_1 & b_0 & b_7 & b_6 & b_5 & b_4 \\ b_4 & b_5 & b_6 & b_7 & b_0 & b_1 & b_2 & b_3 \\ b_5 & b_4 & b_7 & b_6 & b_1 & b_0 & b_3 & b_2 \\ b_6 & b_7 & b_4 & b_5 & b_2 & b_3 & b_0 & b_1 \\ b_7 & b_6 & b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \end{pmatrix} \quad (2)$$

These conditions relative to the above matrix are:

$$\begin{array}{llll} b_0 \neq b_1 & b_2 \neq b_3 & b_4 \neq b_5 & b_6 \neq b_7 \\ b_0 \neq b_3 & b_4 \neq b_7 & b_0 \neq b_7 & \\ b_0 + b_1 = \{01\} & b_3 = b_4 & b_2 = b_7 & b_5 = b_6 \end{array} \quad (3)$$

In order to have optimal construction for the new MDS matrix, we will apply the lemma 1 and lemma 2 so we get:

$$v_1^{8,8} \geq 16 + 8 - 3 = 21 \ , \ c^{8,8} \leq 5 \quad (4)$$

So we choose elements for the new MDS matrix that satisfy both 3 and 4 conditions. The obtained polynomial is $H = had[0x50, 0x51, 0x11, 0x01, 0x01, 0x10, 0x01, 0x11]$ in $GF(2^8)$. By using this polynomial 8×8 matrix is constructed. The MDS matrix with involution property based on above polynomial shown below:

$$\begin{bmatrix} 50 & 51 & 11 & 01 & 01 & 10 & 01 & 11 \\ 51 & 50 & 01 & 11 & 10 & 01 & 11 & 01 \\ 11 & 01 & 50 & 51 & 01 & 11 & 01 & 10 \\ 01 & 11 & 51 & 50 & 11 & 01 & 10 & 01 \\ 01 & 10 & 1 & 11 & 50 & 51 & 11 & 01 \\ 10 & 01 & 11 & 1 & 51 & 50 & 01 & 11 \\ 01 & 11 & 01 & 10 & 11 & 01 & 50 & 51 \\ 11 & 01 & 10 & 01 & 01 & 11 & 51 & 50 \end{bmatrix} \quad (5)$$

In our new system, the data at MDS step is converted into 8×8 matrix and multiplied with the MDS matrix. Each element in the product matrix is the sum of products of elements of one row and one column. In this case, the individual additions and multiplications are performed in $GF(2^8)$. The following Example describes about the multiplication of the state matrix with the above MDS matrix and illustrates the branch number.

$$\begin{bmatrix} 50 & 51 & 11 & 01 & 01 & 10 & 01 & 11 \\ 51 & 50 & 01 & 11 & 10 & 01 & 11 & 01 \\ 11 & 01 & 50 & 51 & 01 & 11 & 01 & 10 \\ 01 & 11 & 51 & 50 & 11 & 01 & 10 & 01 \\ 01 & 10 & 1 & 11 & 50 & 51 & 11 & 01 \\ 10 & 01 & 11 & 1 & 51 & 50 & 01 & 11 \\ 01 & 11 & 01 & 10 & 11 & 01 & 50 & 51 \\ 11 & 01 & 10 & 01 & 01 & 11 & 51 & 50 \end{bmatrix} \times \begin{bmatrix} 00 & 00 & 00 & 00 & 00 & 00 & 6F & 00 \\ 00 & 00 & 00 & 00 & 00 & 74 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 73 \\ 53 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 69 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 6E & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & A4 & 00 & 00 & 00 \\ 00 & 65 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

$$= \begin{bmatrix} 53 & 33 & 69 & 6E & 4A & 2E & 2D & 44 \\ 66 & 56 & 69 & 6E & EE & 5A & 42 & 73 \\ B2 & 65 & 96 & 88 & A4 & 74 & 99 & EB \\ E1 & 65 & FF & E6 & AE & 33 & 6F & 98 \\ 66 & 65 & CC & 13 & EE & 74 & 6F & 37 \\ 35 & 33 & A5 & 7D & A4 & 74 & 6F & 44 \\ 53 & 6A & FF & 6E & 63 & 33 & F6 & 73 \\ 53 & 0F & 69 & 88 & C7 & 47 & 99 & 73 \end{bmatrix} \quad (6)$$

From above we see that our new MDS matrix provides that the maximum branch number of matrix. To make our system has perfect avalanche effect we arrange the plaintext in the following form:

$$\begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\ b_{12} & b_{13} & b_{14} & b_{15} & b_8 & b_9 & b_{10} & b_{11} \\ b_{16} & b_{17} & b_{18} & b_{19} & b_{20} & b_{21} & b_{22} & b_{23} \\ b_{28} & b_{29} & b_{30} & b_{31} & b_{24} & b_{25} & b_{26} & b_{27} \\ b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} & b_{38} & b_{39} \\ b_{44} & b_{45} & b_{46} & b_{47} & b_{40} & b_{41} & b_{42} & b_{43} \\ b_{48} & b_{49} & b_{50} & b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \\ b_{60} & b_{61} & b_{62} & b_{63} & b_{56} & b_{57} & b_{58} & b_{59} \end{bmatrix} \quad (7)$$

B. The Subkeys Generation

The new system key expansion algorithm takes as input a 128-byte key and produces 400 words (1600 bytes). This is sufficient to provide 24-word subkey for each of the 16 rounds of the cipher and 16-word to the output transformation round. In subkey generation process, we try to have maximum avalanche effect between the user key and the ciphertext and to have minimum correlation coefficient. We use MDS matrix to make the subkeys effective against differential cryptanalysis and linear cryptanalysis. The array of subkeys with length 1600 bytes generated by the following mathematica 9 program:

1- The first 128 bytes are the user key bytes.

2- update the rest of bytes by XORing one byte of user key with one updated byte:

$$subkey[128] = subkey[0] \oplus subkey[1];$$

$$For[i = 129, i < 1600, i++,$$

$$subkey[Mod[i - 128, 128]] \oplus subkey[i - 1]$$

$$]$$

The subkeys array arranged into 25 state arrays, and then we apply the MDS matrix to each state array. Finally we convert the byte array to word array to get 400 word subkeys.

C. The Encryption Process

Our proposal is purely block cipher. The input plaintext length is 64 bytes. These bytes are divided into 16 sub-blocks each of 32 bits. The output of this system is also 16 sub-blocks arranged sequentially each of 32 bits. These sub-blocks are combined again to form 64 bytes blocks. The key length is 1024 bits. This key is divided into 400 subkeys each of 32 bits.

Description of a Single Round

The input blocks of 512 bits are divided into 4 sub-blocks each of 128 bits. Each sub-block is divided into 4 32-bit words. Every round has three steps. The round begins with the transformation that combines the four input sub-blocks with 6×4 subkeys, using the 2^{32} modulo addition and $2^{32}-5$ modulo multiplication. The four output sub-blocks are then combined using the XOR operation to form 8 32 bit blocks that are input to the MA structure. The MA structure takes 2×4 subkeys that are input to the MA structure. The output from MA step are arranged into 8×8 state array and multiplied with the MDS matrix. After the 16^{th} round we apply the final transformation that has the same structure as in rounds. The Fig 2 shows the structure of single round.

Key XORed Step

Frequency analysis is the fundamental cryptanalytic technique beside brutal force, threat, blackmail, torture, bribery, etc. The new algorithm tries to defeat the frequency analysis by using the step of "key xored". In this step, we XORed four words (W [4], W [7], W [10] and W [13]) from the first block of data before any round with four subkey words (W [1], W [6], W [11] and W [16]). After we encrypt the first data block the key

xored step is performed at the next data blocks. First, we XORed four words (W [1], W [6], W [11] and W [16]) from pervious output together. The output is 32-bits divided into 4 bytes. The digit value of four bytes represents the number of four subkeys which xored with four words in the input data block (W [4], W [7], W [10] and W [13]). In this step, we choose the first four words from every sub-block because in case of repeated data we need to be sure that the four words are deferent so the xored output can't be zero. We choose the modified words in order to make sure that each input column in matrix for MDS step will change. The overall structure of cipher is shown in Fig.3.

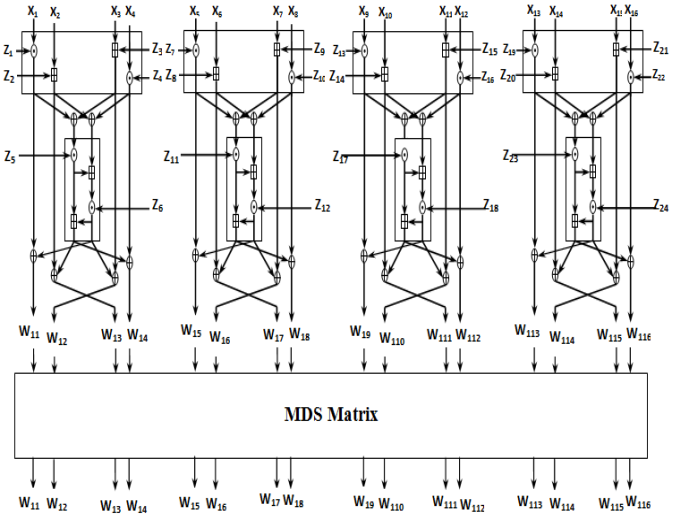


Figure. 2 Single round structure

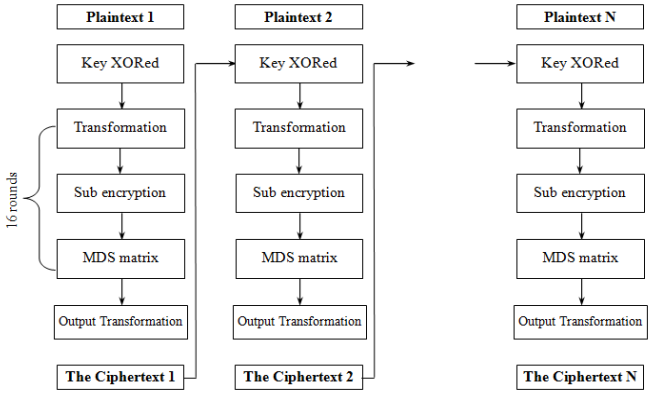


Figure. 3 the new system overall structure

III. Security Analysis

A. *Avalanche Effect*

In cryptography, the **avalanche effect** refers to a desirable property of cryptographic algorithms. The avalanche effect is evident when an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g. half the output bits flip). In the case of quality block ciphers, such a small change in either the key or the plaintext should cause a drastic change in the ciphertext. Constructing a cipher to exhibit a substantial avalanche effect is one of the primary design objectives. The avalanche effect is calculated as:

$$\text{Avalanche Effect} = \frac{\text{No. of flipped in the ciphered text}}{\text{No. of bits in the ciphered text}} \times 100\% \quad (8)$$

In our case, we take two plaintexts, first one is normal message while the second one is repeated zero binary and two plaintexts each one is only 512 in length, flipping one bit from everyone in different positions and calculate the avalanche effect. Then we flip the user key in different positions and calculate the avalanche effect [6]. The following results are obtained after calculating the respective Avalanche Effects.

TABLE.1 AV EFFECT FOR 1 BIT CHANGE IN THE PLAINTEXT

Plaintext	Length of plaintext in bits	Change first bit in plaintext		Change last bit in plaintext		Change middle bit in plaintext	
		IDEA	Proposed	IDEA	Proposed	IDEA	Proposed
Case 1	158720	0.02%	50.1%	0.02%	0.16%	0.03%	25.1%
Case 2	200000	0.02%	50%	0.02%	0.13%	0.01%	25.1%
Case 3	1024	7%	52.5%	6.1%	50.2%	6.6%	54.5%
Case 4	1024	8%	50.6%	5.9%	52.8%	6.3%	51%

TABLE.2 AV EFFECT FOR 1 BIT CHANGE IN THE USER KEY

Plaintext	Length of plaintext in bits	Change first bit in key		Change last bit in key		Change middle bit in key	
		IDEA	Proposed	IDEA	Proposed	IDEA	Proposed
Case 1	158720	49.9%	50.1%	49.9%	50.2%	49.9%	50.2%
Case 2	200000	34.4%	49.9%	43.8%	50%	43.8%	50%
Case 3	1024	34.4%	52.2%	45.3%	50.8%	45.3%	50.8%
Case 4	1024	51%	52.3%	51.2%	54.3%	51.2%	54.3%

The avalanche effect of the proposed algorithm is producing very high as comparison IDEA because in IDEA if only one bit changes, it effects on its data block not all the blocks, while in our proposed system because we XOR the data block with the predetermined subkeys, so if one bit changes it produces different output.

B. *Secret Data Groups*

Considering the secret data used in IDEA, the brute force attack for the key in the case of 128 bit block is $(2^{128} = 3.4 \times 10^{38})$. The brute force attack for the data block in the case of 64 bit block is $(2^{64} = 1.8 \times 10^{19})$.Considering the secret data used in our proposed system, the brute force attack for the key for 1024 bits block is $2^{1024} = 1.8 \times 10^{308}$. The brute force attack for the data block for 512 bits block is $2^{512} = 1.34 \times 10^{154}$.

C. *Language Statistics*

Language redundancy [7] is the greatest problem for any cryptosystem. The cryptanalyst uses the language redundancy to attack cryptosystems ciphertext. If the message is long

enough, the cryptanalyst computes the frequency of each of the characters and consider different number of combinations up to the length of the cryptosystem block. The cryptanalyst will then try to estimate the plaintext from this statistical result. A cryptosystem is considered unbreakable against statistical analysis if its ciphertext has flat distribution. To implement the strength of new system, Figs 4&5 show the plaintext statistics of the used file. The ciphertext statistics of IDEA and new proposed system are plotted in Figs 6 to 9.

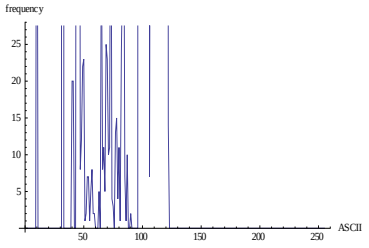


Figure.4 Plaintext statistics

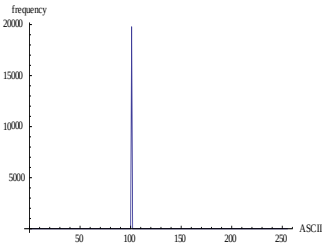


Figure.5Repeated plaintext statistics

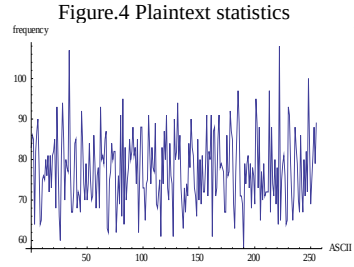


Figure.6 Proposed ciphertext statistics

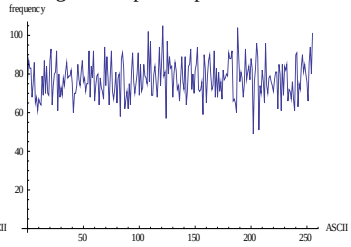


Figure.7 IDEA ciphertext statistics

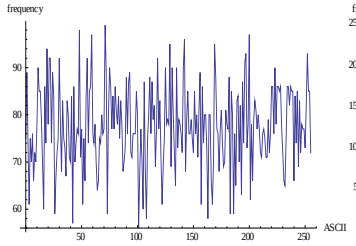


Figure8 Proposed ciphertext statistics of character "e" message

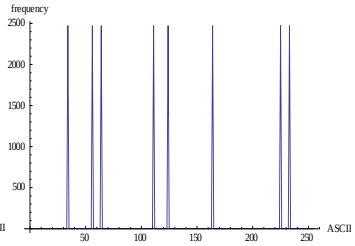


Figure.9 IDEA ciphertext statistics of character "e" message

D. *NIST Statistical Suite*

The National Institute of Standards and Technology (NIST) [8] develops a Test Suite as a statistical package consisting of 16 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based Cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests. The average values of the statistical tests for both algorithms were given in Table 3.

TABLE 3 LKES VS. LUCIFER STATISTICAL TESTS

Test name \ Algorithm	Proposed system		IDEA	
Frequency (Monobit) Test	100%	Pass	100%	Pass
Frequency Test within a Block	100%	Pass	99%	Pass
Runs Test	100%	Pass	100%	Pass
the Longest Run of Ones in a Block Test	100%	Pass	100%	Pass
Binary Matrix Rank Test	100%	Pass	98%	Failed
Discrete Fourier Transform Test	100%	Pass	100%	Pass
Non-overlapping Template Matching Test	100%	Pass	100%	Pass
Overlapping Template Matching Test	99%	Pass	100%	Pass
Maurer's "Universal Statistical" Test	100%	Pass	100%	Pass
Lempel-Ziv Compression Test	100%	Pass	98%	Failed
Linear Complexity Test	100%	Pass	99%	Pass
Serial Test	100%	Pass	98%	Failed
Approximate Entropy Test	100%	Pass	100%	Pass
Cumulative Sums (Cusum) Test	100%	Pass	100%	Pass
Random Excursions Test	99%	Pass	100%	Pass
Random Excursions Variant Test(α = 0.05)	96%	Pass	92%	Failed

IV. Conclusion

In this paper, we introduce a new cipher based on IDEA cryptosystem. We have improved the security of IDEA by increasing the size of data block to 512 bits and the size of key to 1024 bits. We use modulo the prime field $2^{32}-5$ multiplication to increase the strength of the proposal cipher. We introduce a new optimal diffusion 8x8 MDS matrix. This new matrix meets all requirements of optimum design. We use the MDS matrix in the key expansion procedure to make it strong against the known attacks. In our proposed system if we change a few bits in the plaintext or the user key it cause more than half of the ciphertext to be change. Finally, our proposal is rigid to withstand the well-known methods of brute-force.

References

[1] Mediacrypt AG, The IDEA Block Cipher, submission to the NESSIE Project.

[2] Behrouz A.Forouzan "Cryptography and network security "TATA-Mcgraw hill publication 2007 edition.

[3] Junod, P., Vaudenay, S.: Perfect Diffusion Primitives for Block Ciphers: Building Efficient MDS Matrices. In: Selected Areas in Cryptography (2004).

[4] F. MacWilliams and N. Sloane. The theory of error-correcting codes. North-Holland, 1977.

[5] Daesung Kwon, Soo Hak Sung, Jung Hwan Song and Sangwoo Park," Design of Block Ciphers and Coding Theory". Trends in Mathematics, Information Center for Mathematical Sciences Volume 8, Number 1, June, 2005, Pages 13-20.

[6] Amish Kumar, "effective implementation and avalanche effect of AES", International Journal of Security, Privacy and Trust Management (IJSPTM), Vol. 1, No 3/4, August 2012.

[7] Bruce Schneier, "Applied Cryptography, Protocols, Algorithms, and Source Code in C" Wiley Computer Publishing, Second Edition, John Wiley & Sons, Inc.

[8] NIST, "A Statistical Test Suite for Random and Pseudorandom Generators for Cryptographic Applications", NIST Special Publication 800-22, 2003.