ENHANCED IDEA ALGORITHM FOR STRONG ENCRYPTION BASED ON EFFICIENT STRONG ROTOR BANKS

H. Elkamchouchi[1], Fatma Ahmed[2]
[1, 2] Electrical Engineering Department, Faculty of Engineering, Alexandria University
[1] Helkamchouchi@ieee.org
[2] moonyally@yahoo.com

ABSTRACT

Information security becomes an important issue of the communication networks. In this paper we propose 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 17. It uses modulo 2^{32} addition and thirty-two bits XORING are used followed by modulo $2^{32}-5$ multiplication which is prime number to increase the field in multiplication operation. It uses a new efficient and strong rotor bank which provides best resistance against linear and differential cryptanalysis, also provides better resistance against algebraic attack because it is implemented using Kasami Exponent function one of APN (Almost Perfect Nonlinear) function. The rotor banks rotate by irregular step and we use it to generate the subkeys which give us high diffusion and confusion. In this system, we 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, APN, Kasami exponent, rotor, frequency analysis.

1 INTRODUCTION

1.1 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.

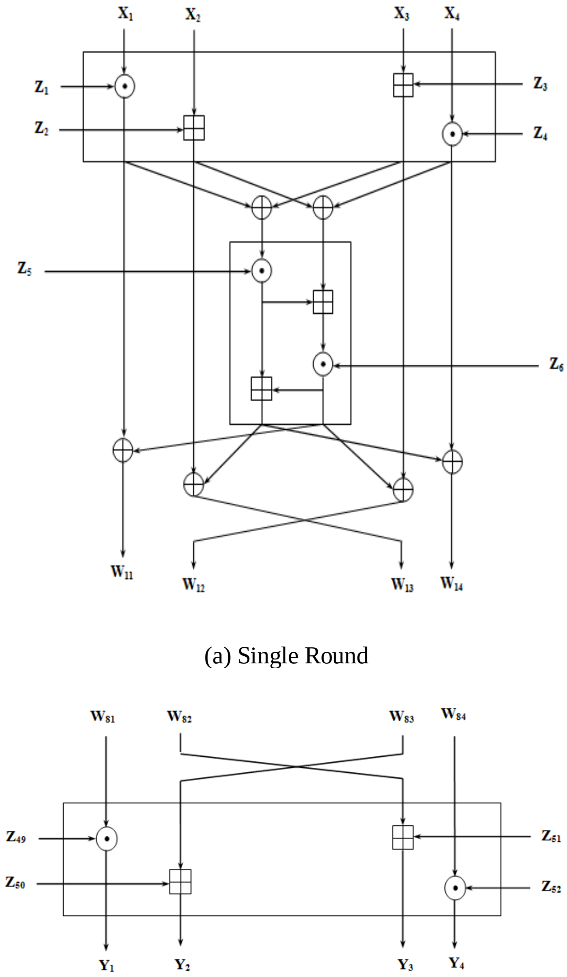
1.2 Almost Perfect Nonlinear

Let $L=\mathbb{F}_q$ with $q=2^n$ for some positive integer n. A function $F:L\to L$ is said to be "almost perfect nonlinear" APN [2] on L if the number of solutions in L of the equation:

ISBN: 978-0-9853483-6-6 ©2013 SDIWC

$$\delta(F)=\max_{a\neq 0,b}\left\{x\in F_2^n, F(x+a)+F(x)=b\right\} \quad (1)$$

$\delta(F)$ is at most 2, for all $a,b\in L, a\neq 0$. Equivalently, F is APN if the set $\left\{x\in F_2^n, F(x+a)+F(x)\right\}$ has size at least 2^{n-1} for each $a\in L*$. Because L has characteristic 2, the number of solutions to the above equation must be an even number, for any function F on L .Note that APN functions have the best resistance against differential cryptanalysis.



(a) Single Round

(b) The output transformation

Figure 1 The overall structure of IDEA

1.2 Kasami Exponent

A Kasami exponent, one of APN functions, is defined as:

Let t be an integer such that $2\leq t\leq n/2$. Let us define the functions on $\mathbb{F}_{2^n}$ where $n=2m+1$: $d=2^{2t}-2^t+1$ with $\gcd(t,n)=s$ and n/s is odd, Then $\delta(F)\in\{0,2^s\}$ so $s=1$. From [3] we can see that the power functions with Kasami exponents have slightly better resistance against algebraic attacks.

2. 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 17. In the new system we: proposed new efficient rotor banks and we introduce new rotated step for rotor to resist the frequency analysis attack. Also we permute the data after each round.

2.1 The New Efficient Rotor Banks

In this paper we introduce new rotor banks implemented using the Kasami exponent function. The rotor banks consist of two rotors first one generates on $GF(2^7)$ and the second rotor generates on $GF(2^9)$. We use n odd to satisfy the condition n/s where $s=1$. This condition guarantees that F is a permutation. Each rotor has four cylinders in different irreducible polynomials. When we search for irreducible polynomials in $GF(2^7)$ and $GF(2^9)$ we found that several irreducible polynomials doesn't generate a permutation. Examples of these irreducible polynomials are listed in Table 1.

Table 1. Examples of irreducible polynomials doesn't generate permutation

irreducible polynomials on $GF(2^7)$	irreducible polynomials on $GF(2^9)$
$1+x+x^3+x^7$	$1+x^2+x^3+x^7+x^9$
$1+x^2+x^4+x^7$	$1+x+x^4+x^7+x^9$
$1+x^3+x^5+x^6+x^7$	$1+x^3+x^4+x^5+x^9$
$1+x^2+x^4+x^5+x^7$	$1+x+x^3+x^4+x^5+x^9$

$1+x^2+x^3+x^6+x^7$	$1+x^2+x^3+x^4+x^6+x^7+x^9$
$1+x+x^4+x^5+x^7$	$1+x+x^3+x^4+x^5+x^6+x^7+x^9$
$1+x^2+x^3+x^5+x^7$	$1+x^2+x^3+x^4+x^5+x^6+x^7+x^8+x^9$

The rotor banks ware attempt in the following way:

1- ***Initialize the rotor banks:*** the first bank of rotor has four cylinders; each cylinder initialized using range 0 to 127 while the four cylinders of the second bank initialized using range from 0 to 511.

2- ***Map each element in the rotor banks using the Kasami function:*** each element in the first bank of rotor mapping using the Kasami exponent in$GF(2^7)$ for the following polynomial:

   a- Rot1: $1+x^6+x^7$

   b- Rot2: $1+x+x^2+x^5+x^7$

   c- Rot3: $1+x^3+x^7$

   d- Rot4: $1+x+x^7$

   For the second bank, each element mapping using the Kasami exponent in $GF(2^9)$ for the following polynomial:

   a- Rott1: $1+x^2+x^4+x^8+x^9$

   b- Rott2: $1+x^2+x^4+x^7+x^9$

   c- Rott3: $1+x^2+x^5+x^6+x^9$

   d- Rott4: $1+x^3+x^5+x^6+x^9$

3- ***Arrange the rotor banks:*** in order to get low correlation between input and output $\approx 0.4\times 10^{-3}$ we arrange the cylinders in rotor banks as following:

   Rotor banks= {Rot1, Rot2, Rot4, Rot3, Rott3, Rott1, Rott4, Rott2}

The new rotor banks will be rotated after each block is enciphered. The second bank will be rotated in a manner similar to "scrambled car odometer". The first bank of rotor will be rotated in irregular step based on the output from it and the output from the second bank. First step is XORing the output from the first bank with the output from the second bank. The resulting output is reduced to $GF(2)$ by applying mod operation$(1+x^2)$. This output is causing the associated cylinder in the first bank to turn after a letter is encrypted. This step resists the frequency analysis attack and the well known methods of brute-force attack [4] because we make sure that even for repeated data blocks, the ciphertext will not be repeated blocks.

2.2 Subkeys Generation

The new system has key expansion algorithm takes as input a 32-word (128-byte) key and produces $[6\times 4\times 17+4\times 4]=424$ words. This is sufficient to provide 6×4-word (96 bytes) subkeys for each of the 17 rounds of the cipher and 4×4-word in the output transformation. In subkey generation process, we try to have maximum confusion effect between the user key and the ciphertext and to have minimum correlation coefficient. The subkeys are generated by using the rotor banks. First, we initialize the subkeys array 1096-bits from user key by using the following mathematica.9 sub-program:

```
 Keyr = Flatten[Key];
 subKey = {0};
 subKey = PadRight[subKey, 905];
 For[i = 1, i < Length[subKey], i++,
        subKey[[i]] = Take[Keyr, 15];
         Keyr = RotateLeft[Keyr, 15]
    ]
subKey[[905]] = Take[Keyr, 8];
subKey = Partition[Flatten[subKey], 16];
```

We apply the rotor banks to the pervious output array. Using rotor banks in subkeys generation make the subkeys resist the known cryptanalytic attacks. After each block is encrypted, the both two banks will rotate in a manner similar to "scrambled car odometer".

2.3 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

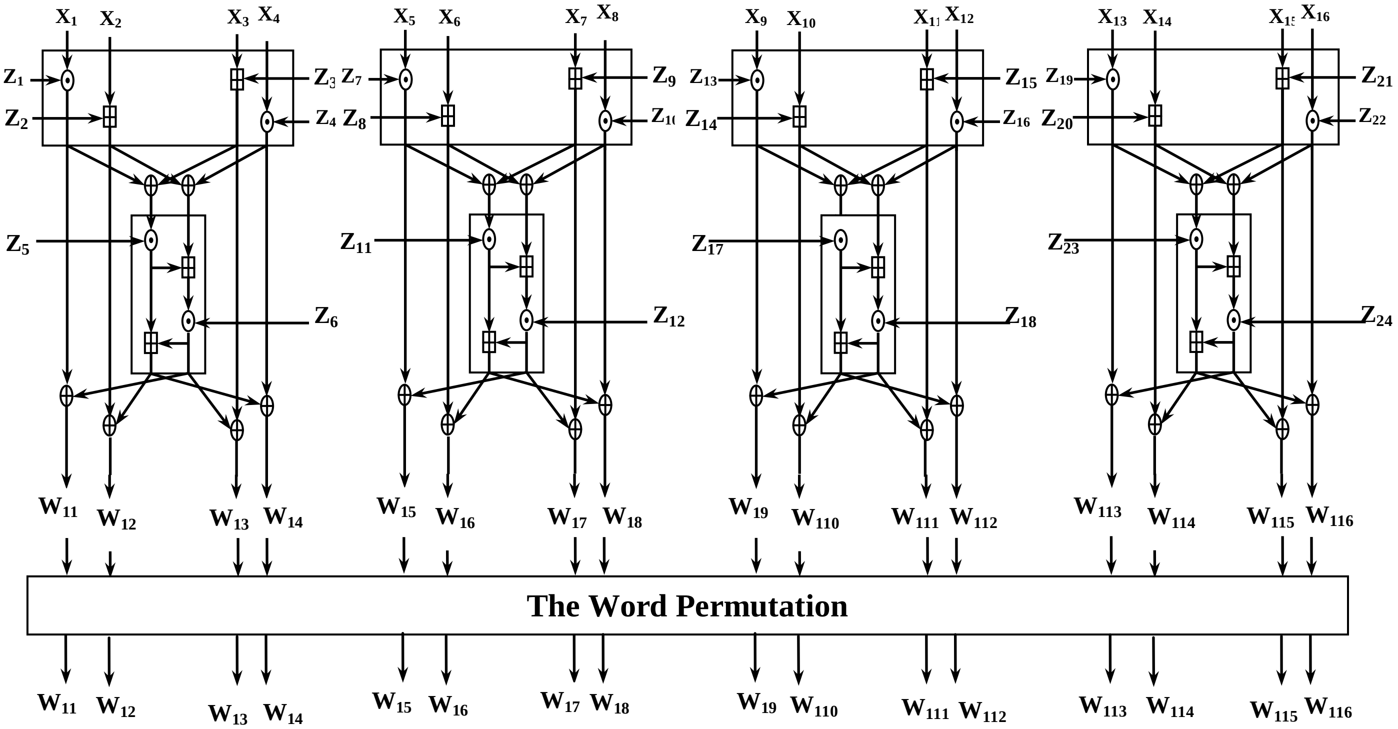
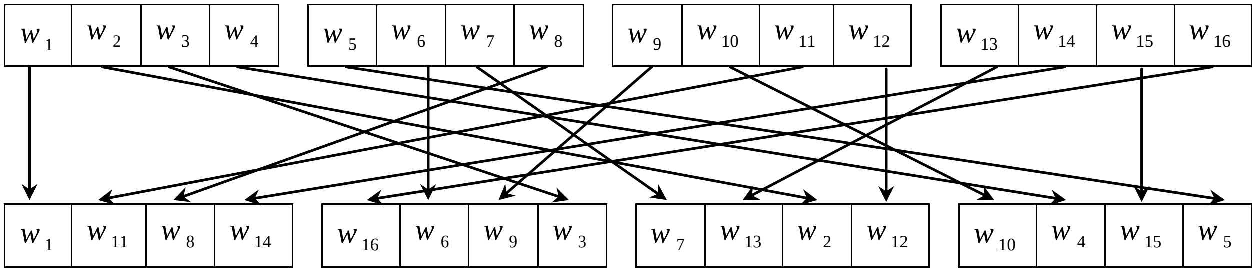


Figure 2. Single round structure



w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11 w12 w13 w14 w15 w16
w1 w11 w8 w14 w16 w6 w9 w3 w7 w13 w2 w12 w10 w4 w15 w5

Figure 3. Word permutation

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 is permuted using the word permutation. In the middle of round nine, we apply the rotor step so we maintain the symmetry for encryption and decryption because the algorithm of encryption is the algorithm of decryption. In the rotor step, the input data is divided into sub-blocks each one is sixteen bits. Every sub-block is divided into two parts; the first part is nine bits which encrypted using the second bank of rotor and the second part is seven bits which encrypted using the first bank of rotor. After the rotor step the data will be flatten and divided into 4 sub-blocks each of 128 bits. After the 17^{th} round we apply the final transformation that has the same structure as in rounds. The Fig 2 shows the structure of single round.

2.4 Permutation

In order to have high diffusion between plaintext and ciphertext we need to make sure that every byte in plaintext will effect on all other bytes. The permutation is illustrates in the Fig. 3.This permutation was tested on a random sample of the messages, in each time it guarantees the high diffusion between the input and the output because we swap between three words in each sub-block to one word in other three sub blocks. The overall structure of cipher is shown in Fig.4.

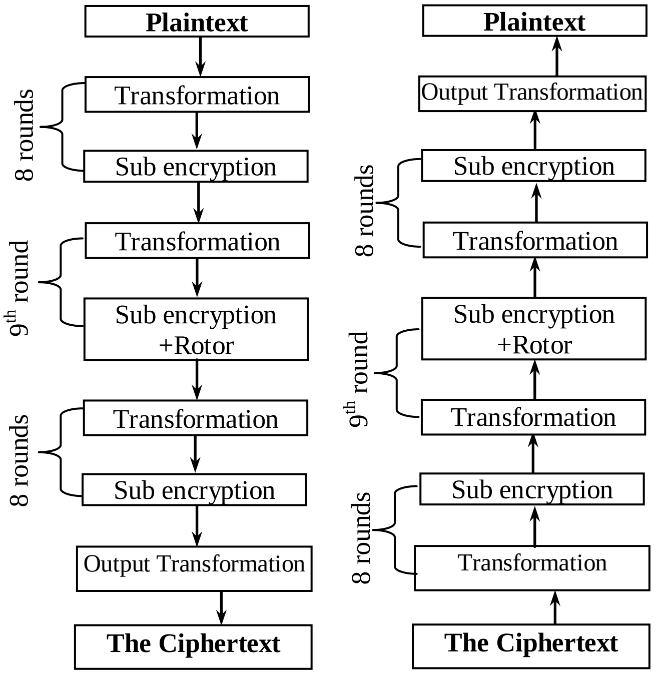


Figure. 4 New system encryption and decryption

3. SECURITY ANALYSIS

3.1 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:

$$\text{Avalanche Effect}=\frac{\text{No. of flipped in the ciphered text}}{\text{No. of bits in the ciphered text}}\times 100\% \quad (2)$$

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 [5]. The following results are obtained after calculating the respective Avalanche Effects.

Table.**2**avalanche effect for 1 bit change in the 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
158720	0.02%	50.1%	0.02%	0.16%	0.03%	25.1%
200000	0.02%	50.2%	0.02%	0.14%	0.01%	25.1%
1024	7%	53.5%	6.1%	54.3%	6.6%	54.5%
1024	8%	52.6%	5.9%	52.8%	6.3%	52.5%

Table.3 avalanche effect for one bit change in the user key

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
158720	49.9%	50.1%	49.9%	50.2%	49.9%	50.3%
200000	34.4%	50%	43.8%	50.2%	43.8%	50.3%
1024	34.4%	52.2%	45.3%	51.5%	45.3%	51.6%
1024	51%	52.4%	51.2%	54.3%	51.2%	51.5%

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 permute the data blocks together, so if one bit changes it produces different output.

3.2 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 $(2^{128}=3.4\times10^{38})$. The brute force attack for the data block in the case of 64 bit block is $(2^{64}=1.8\times10^{19})$ .Considering the secret data used in our proposed system, the brute force attack for the key for 1024 bits block is $2^{1024}=1.8\times10^{308}$. The brute force attack for the data block for 512 bits block is $2^{512}=1.34\times10^{154}$.

3.3 Language Statistics

Language redundancy [6] 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 5&6 show the plaintext statistics of the used file. The ciphertext statistics of IDEA and new proposed system are plotted in Figs 7 to 10.

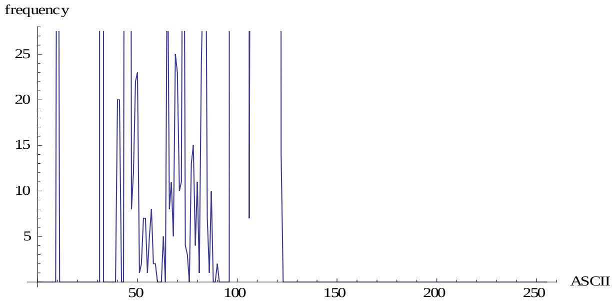


Figure.5 Plaintext statistics

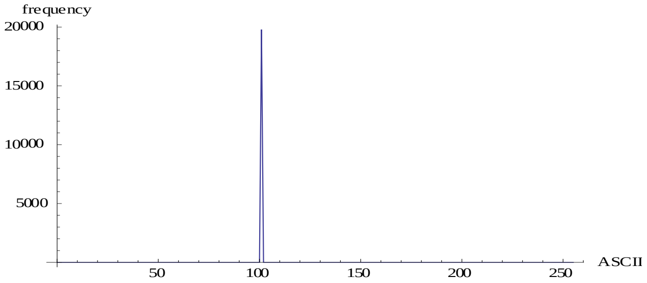


Figure 6 plaintext statistics

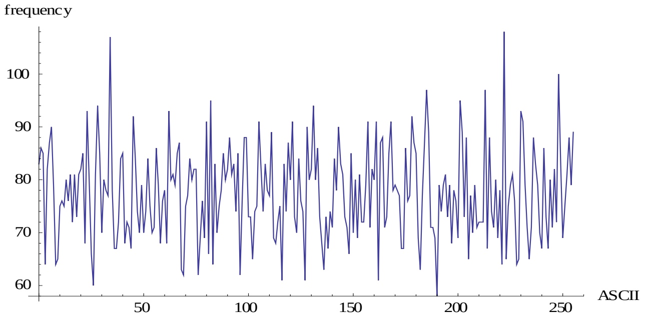


Figure.7 Proposed ciphertext statistics

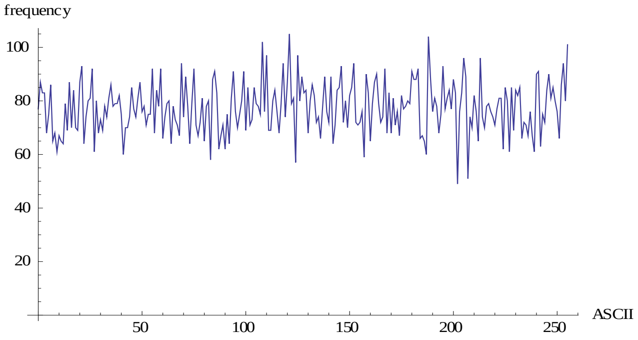


Figure. 8 IDEA ciphertext statistics

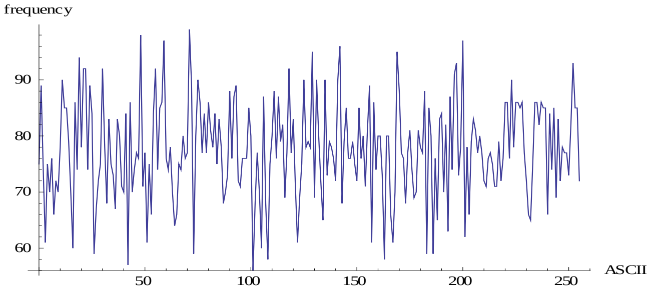


Figure 9 Proposed ciphertext statistics of character "e" message

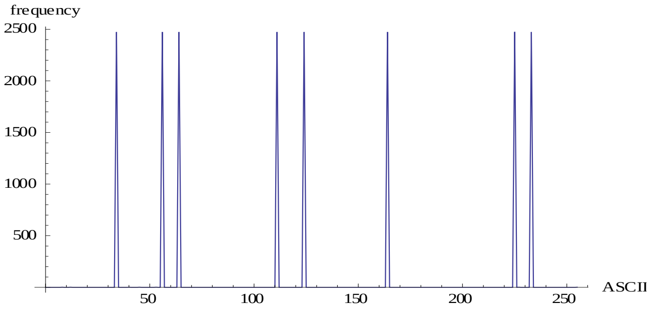


Figure.10 IDEA ciphertext statistics of character "e" message

3.4 NIST Statistical Suite

The National Institute of Standards and Technology (NIST) [7] 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 4.

Table.4 proposed system vs. Idea statistical tests

Test name **Algorithm**	Proposed system		IDEA	
Frequency (Monobit) Test	100%	Pass	99%	Pass
Frequency Test within a Block	99%	Pass	96%	Failed
Runs Test	100%	Pass	100%	Pass
the Longest Run of Ones	100%	Pass	100%	Pass
Binary Matrix Rank Test	100%	Pass	99%	Pass
D Fourier Transform Test	100%	Pass	100%	Pass
Non-overlap Template Match	100%	Pass	100%	Pass
Overlap Template Match Test	100%	Pass	100%	Pass
Maurer's Universal Statistical	100%	Pass	100%	Pass
Lempel-Ziv Compression Test	99%	Pass	99%	Pass
Linear Complexity Test	100%	Pass	98%	Failed
Serial Test	99%	Pass	98%	Failed
Approximate Entropy Test	100%	Pass	100%	Pass
Cumulative Sums Test	100%	Pass	100%	Pass
Random Excursions Test	100%	Pass	98%	Failed
Random Excursions Variant Test($\alpha = 0.05$)	96%	Pass	93%	Failed

4. 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 efficient rotor banks using Kasami exponent. We use the rotor banks 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.

5 REFERENCES

1. Mediacrypt AG, The IDEA Block Cipher, submission to the NESSIE Project.
2. Aubry, Y., McGuire, G., Rodier, F.: A few more functions that are not APN infinitely often. In: McGuire, G., et al. (eds.) Finite Fields: Theory and Applications, Ninth International conference Finite Fields and Applications, Contemporary Math. n° 518, pp. 23–31. AMS, Providence (2010).
3. Jung Hee Cheon, Dong Hoon Lee, Resistance of S-Boxes against Algebraic Attacks, pp. 83 - 94, FSE 2004.
4. Mohd Zaid Waqiyuddin Mohd Zulkifli "Attacks on Cryptography"April 2008.
5. 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.
6. Bruce Schneier, "Applied Cryptography, Protocols, Algorithms, and Source Code in C" Wiley Computer Publishing, Second Edition, John Wiley & Sons, Inc.
7. NIST, "A Statistical Test Suite for Random and Pseudorandom Generators for Cryptographic Applications", NIST Special Publication 800-22, 2003.