[Ganesh, 5(7): July 2018]
DOI: 10.5281/zenodo.1310796

ISSN 2348 – 8034
Impact Factor- 4.022

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

AN ENHANCED HASH BASED CRYPTOGRAPHIC ALGORITHM

S. Sai Ganesh[*1], Dr. G.S.N.Murthy[2], T.Veerraju[3] & B.R.S.S.Raju[4]

[*1] Asst Professor, Dept of CSE, Aditya College of Engineering, Surampalem, INDIA

[2] Professor, Dept of CSE, Aditya College of Engineering, Surampalem, INDIA

(Orcid Id: 0000-0003-2967-9347)

[3] Associate Professor, Dept of CSE, Aditya College of Engineering, Surampalem, INDIA

[4] Asst Professor, Dept of CSE, Aditya Engineering College, Surampalem, INDIA

ABSTRACT

We are having various Conventional Encryption/Decryption techniques for encrypting the data to protect security while transferring the data in a networked environment. Now-a-days, as threats become more widespread, attackers are getting smarter and the infrastructure required securing. Hence, it is necessary to modify the structure of an existing algorithm or introducing a new algorithm to enhance the security. In this study, we are modifying the functionality of the DES Algorithm to enhance its security by consistently maintaining the integrity and authenticity of the message. In conventional encryption, Data Encryption Standard (DES) algorithm is a block cipher algorithm which uses a 64-bit input and a 56-bit key, which is not much secure and can be susceptible by Brute force attack by trying a possible 2^{56} Keys. The Enhanced DES (EDES) which uses a Hash function to produce a haphazard finger mark for every block of the Plaintext. This finger mark is used to create the secure bits to produce round seeds are used to select secure S-boxes only for each round in the encryption process. In this not only the randomness of bits were generated for S-Box, prior to that an expansion function is also modified to strengthen the proposed algorithm. The Main advantage of this enhanced method reduces the processing time and enhances the security by increasing complexity, confusion and diffusion of this algorithm which leads more security during the data transmission because we are applying various transpositions. The New algorithm designed is referred to as EDES-H. The EDES-H performance is analyzed and compared with the DES, DESX, 2DES, 3DES and AES, using a number of metrics including Time for Encryption/Decryption, Simple regression and Correlation analysis, Hamming distance and Disparity of cipher.

Keywords: *DES, S-Boxes, P-Box, Encryption Algorithm, EDES-H, Disparity of cipher.*

I. INTRODUCTION

Now-a-days, IT sector focuses on information security with the help of various encryption techniques. They can be classified into Symmetric Key Encryption (SKE) and Asymmetric key Encryption (ASKE) algorithms. In SKE, to encrypt and decrypt the data we use a single key. In ASKE, two separate but related keys are used to encrypt and decrypt the data [1]. The development of the internet technology allows easier access to users' sensitive information. The unauthorized entry into the resource, amendment of information and destruction or obstruction of the useful information causes a momentous damage to the belongings of the information. The useful and important information should be encrypted during transmission over the networks which are prone to access by the unauthorized users. Hence it is required the very fast and much efficient encryption processes that can be capable of protecting the users privacy. The disturbance and impediments which are due to the particular encryption algorithms will disturb the processes.

The encryption algorithm needs to take so many constraints while we need to design the structure like the efficiency, speed etc. Along with that it has to support that the information can be passed through the multiple components without any loss/obstruction. Security of the data plays an Essential role While the transmission of the data through the network. We have lot of techniques to protect the data where encryption plays the vital role in the Security of the information system. The main problem with the DES algorithm is its operation of S-Boxes, which is more and more vulnerable to the linear and differential cryptanalysis attacks. Before that, the Expansion function must be also

[Ganesh, 5(7): July 2018] ISSN 2348 – 8034
DOI: 10.5281/zenodo.1310796 Impact Factor- 4.022

critical to defend such type of attacks to defend. Otherwise, the attackers concentrate more and more on the structure of the algorithm along with the content in particular [2].

The correspondence between the plaintext and the cipher text is analyzed [3] by the attackers. The major level of linkage between the plaintext and cipher text becomes ease to the attacker to involve in the process of encryption directly or to guess the key. By trying various alterations makes ease to find the relationship between the plaintext and the cipher text. Whenever the attacker can get these relationships or guessing the key, then immediately focus on the design of the substitution process and the permutation functions. Hence it is necessary to introduce some new amendments to the algorithm by modifying structure and alteration to the principles of the designing the S-Box to decrease of finding of the correspondence between the cipher text and the plaintext because in some cases the S-box creates same output with two chosen inputs.

A method is introduced which is a key-dependent transformation that can rearrange the original S-Boxes in different ways using the last 5 bits of the key in DES [4]. This method strengthens the DES by evading of weakening the some of the S-Boxes in respect to the linear and differential cryptanalysis. But this method not addressed the problem of repeating the same set of S-Boxes in various rounds. Another method proposed by modifying the S-Boxes and made more secure the DES with rotating masks and secured S-boxes at hardware level [5]. In this method, after each and every encryption round the masks are rotated and gives two more extra inputs which are called as position and count. These two extra inputs are used to point the corresponding entry in the S-box to change on the Position value and the round of the present encryption.

The DES algorithm with the Hashed component is a variant to the DES for each block of the plain Text to produce an arbitrary Hash code used as a finger mark. This finger mark is used by the secure bit generator along with the key to produce the secure-bit to generate round seeds which are used in S-boxes for each round in the process of encryption [6]. This work is enhanced by the same with increasing the Randomness in DES Ciphers Produced with two variants [7]. Even though it is a worthy to some extent, but for the performance of various considerations, it has not satisfied all the constraints. In this work it is defined two variants of the DES named DDES and HDES. In some aspects the DDES outperforms HDES. If we take the encryption time as a parameter for consideration, the HDES outperforms the DDES. In both of the variants DDES and HDES, various components are taken. The components are increased the complexity of the algorithm and the time taken for encryption is very high. Hence, we need to remove some of the components to improve the performance. Hence, to satisfy all the performance considerations and to increase the efficiency, we are introducing an enhanced DES algorithm with the hash component (EDES-H).

In this work, we proposed to modify the structure of an Expansion function and further the Input which is sent to the S-Box for each round is varied to strengthen. This paper investigates the DES which is designed to provide the safety to different applications like the Speed of the processing and intensity of sensitivity information. To achieve the above goals of the security system, the encryption algorithms must provide an adequate strength with the elevated security which is implemented in an accepted speed limitation. To improve the performance evaluation on the existing DES encryption algorithm, several approaches are proposed. In this, we are enhancing those approaches by modifying the Expansion function and improving the design of the S-Boxes.

The remaining part of this paper is organized as: Sections II & III describes the EDES-H overview and its working principles and its components. Section IV evaluates the performance of the proposed method against the well-known DES and some other counterparts. Finally, Section V had given the conclusion and future scope of the work.

II. EDES-H OVERVIEW

EDES-H uses some random bits as secure bits which are derived from the plaintext using the Hash function along with the Key. In each round of the process, in encryption, these secure bits are made to the proper arrangements of secured S-boxes which are extracted from the pool. To make the proper arrangement of automatically generated S-boxes dynamically, the plaintext is sent to the hashing function to produce an inimitable finger mark that is used to

[Ganesh, 5(7): July 2018] ISSN 2348 – 8034
DOI: 10.5281/zenodo.1310796 Impact Factor- 4.022

engender the secured bit. The secure bits which are generated through the Generator are entrenched within the generated cipher text, which will be used in the process of decryption at the receiver side.

EDES-H generates only secure arrangements of S-boxes. In EDES-H, the secured bits are generated through the secure bit generator by taking input from the hashed component of the plain text and the Key.

The Components of EDES-H are described as follows: Hashing Algorithm, secure bit generator, Key scheduler, and insertion of S-Bits, Filter. The hashing Algorithm is used to produce a finger mark by taking the plaintext as the input. The secure bit generator takes the Hash code and XOR it with the encryption key to construct the secure bit. The S-boxes takes the round key to dynamic generation of S-boxes arrangements for each round. The Insertion of S-bits is used to produce a cipher text which is embedded with the cipher text. The filtering component is used during the decryption process to extract the seed from the cipher text to perform the process of the decryption. In this variant not only concentrated on the S-Box, but also modifies the expansion function prior to giving input to the S-Box to strengthen the structure. Detailed descriptions of the components and pseudo code are provided in Section III.

The following steps clearly illustrate the encryption process in EDES-H which is depicted in Fig. 2. A pseudo code that helps in understanding the operation of EDES-H is given further.

1. The plaintext is given as the input to the hashing algorithm. The Hash code which is produced from the Hash function is send to the Secure-bit generator.
2. The plaintext is split into two Halves (L and R), each of which an a size of 32 bits
3. The key scheduler generates a round key, which will be XOR ed with the expanded R of size 48 bits. The resulting text is fed into the S-boxes and then permuted. The resulting 32 bits text are then XOR with L
4. A new text is produced of size 64 bits by joining the text from step 3 with R as the, R will become as the left part
5. Now, the L and R goes through the 16 rounds of the encryption process, each of which it uses a different sub key:
   a. From the 64-bit key, a different 48-bit Sub-keys are generated using Key Transformation.
   b. Using the Expansion Permutation, the R is expanded from 32 bits to 48 bits, by using 4x8 matrices.

Expansion function is modified as by taking the 4x8 matrix instead of 8x4 matrix which leads to more confusion of the function, resulting to improve the complexity of the Algorithm also. The expanded positions are shown with Red Color in the Fig 1.

31	32	1	2	3	4	5	6	7	8	9	10
7	8	9	10	11	12	13	14	15	16	17	18
15	16	17	18	19	20	21	22	23	24	25	26
23	24	25	26	27	28	29	30	31	32	0	1

Fig.1: Expansion function

In the modified Expansion function, we padded two columns to both the left side and right side of the actual function. As the cyclic successive bits are padded to the right side of the matrix and anti-successive bits are padded to the left side of the matrix. Then resultant matrix becomes the 4x12 matrix, which increases the complexity of the function results the complexity to the algorithm also.

c. Now, the 48-bit key is XOR ed with 48-bit R and obtained result is carried to the next step.
d. Using the S-box operation, it produces the 32-bit output by taking the 48- bit input using eight S-Boxes.

[Ganesh, 5(7): July 2018] ISSN 2348 – 8034
DOI: 10.5281/zenodo.1310796 Impact Factor- 4.022

e. The resultant 32 bits are permuted using the P-Box.
f. The permuted output of the P-Box is XOR ed with the L of 32 bits and produce R of the result
g. The resultant L is swapped with the R of initial permuted text.

6. The text generated upon completion of the 16 rounds symbolizes the produced cipher text.
7. At last, the insertions of S-bits are implant with the cipher text.

III. COMPONENTS OF EDES-H

Hashing Components:
The Principal initiative of using the hashing algorithm in this DES variant is to create a finger mark of the plaintext. The significance of using the Hashing algorithm which it produces the unique hash code for the given Plain text at Initial round. The hash function takes the input as 64 bits plaintext. The hash code generated by the hash algorithm must not be the same for the different plaintexts and it has to satisfy the Collision-resistant principles of the Hash Algorithm. While designing this architecture of the EDES-H Algorithm, it needs to consider the processing time of the Hashing algorithm which affects the performance of the whole component. The Considered Hash algorithm must not only fast and also easy to reduce the overhead included to use in this architecture.

The processing speed of the various cryptographic hash algorithms such as various versions of SHA and MD, other Hash algorithms like HAVAL, Tiger, Whirlpool and RadioGatun are analyzed [9]. While analyzing the Hash Algorithms, it is taken into consideration of all the factors like Compiler platform, S/W and H/W environments, Compression and round functions, No of steps used, Mathematical functions and the Size of the constant used etc., The analysis resulted that SHA-1 version is better than all other Versions of the SHA and also proved to be better than the MD4, MD5 in terms of processing speed.

Hence, in this work, EDES-H selects SHA-1 as a hash algorithm by taking the considerations into the trade-offs between the Speed of the hashing Algorithm and also the collision resistance. HDES uses the 16 rounds in the process of encryption like the DES because the possibility of occurrence of collision has to be reduced.

Secure Bit Generator: The important component in the EDES-H is the Secure bit Generator. The Secure Bit generator takes the input from the Hash Algorithm and is XOR ed it with the encryption key. The resultant bits are used as a Secure(S) of 8 bits. The major idea behind the use of secure bits is to get a random and unique string that has one to more association with the Key and the plaintext. Hence, we have to fabricate an exclusive secure bit to be fed into the cipher text. Because of the distinctiveness of the Hash algorithm, a small variation in the plaintext may produce a drastic secure bit that would be random due to the XOR operation. This change will occur because of the characteristics of the XOR operation, when a fixed distribution string is XOR ed with a uniform distribution string, the resulting string would follow the uniform distribution. In order to reduce the resulted 64 bits into 8 bits, to create the secure bits, the 64-bit are divided into 8 groups from left to right. To make it easy in the process of reducing by maintaining the randomness, the preliminary bit of every group will correspond to a bit in the secure bit.

[Ganesh, 5(7): July 2018]
DOI: 10.5281/zenodo.1310796

ISSN 2348 – 8034
Impact Factor- 4.022

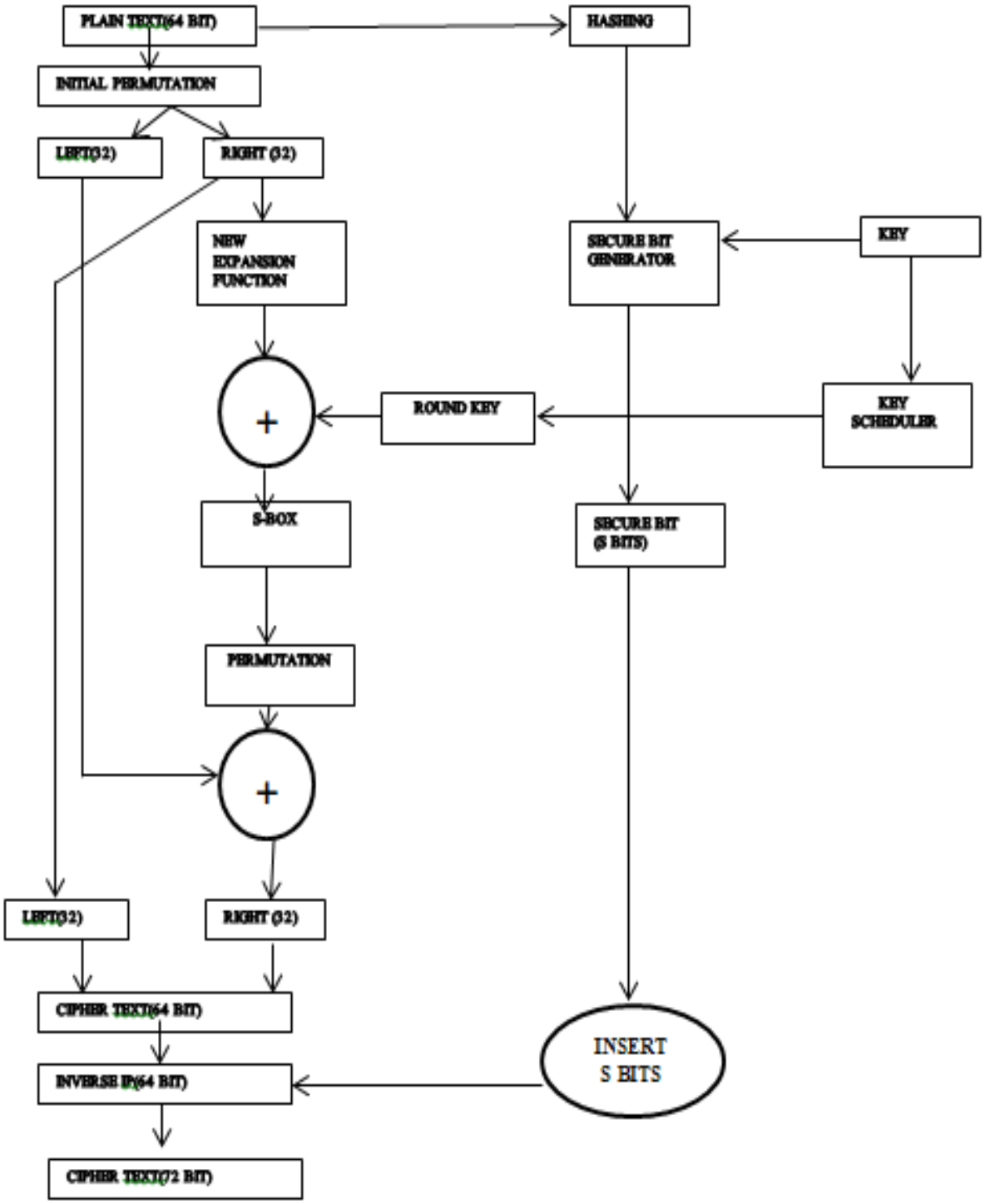

Fig.2: Structure of the EDES-H Algorithm

ISSN 2348 – 8034
Impact Factor- 4.022

Pseudo code:

```
EDES-H ENCRYPTION PROCESS ()
//INPUT: Plaint Text(PT) (64 bit) and Key(64 bit)
//OUTPUT: Cipher Text(CT) (72 bit)
{
        H=Hash(PT);
        IPT=IntialPermutation(PT);
Secure-Bit=SecureBitGenerator (H, Key);
        For round=1 to 16 {
Divide the IPT into 2 Halves of each 32 bit.ie Left (32 bit) and Right (32 bit);
                T=Right;
                ET=New Expansion(Right);
                RK=KeyScHedular (Key);
                ET=ET XoR RK;
                Right=Permutation (ET) XoR Left;
                 Left=T;
                  P=Concatenate (Left, Right);}
           P=InverseInitialPermutation (P);
           CT=InsertSecurebit(P,SecureBit);
}
```

IV. EVALUATING THE PERFORMANCE

Unpredictable random bits must be produced as cipher text by the Encryption algorithms. But in some cases, to decode the cipher text, the attacker tries to analyze, which are not much strengthened in terms of superior randomness. When the cipher text is unable to guess and when the chances of zeros and ones in the cipher text should not be imaginable by the attackers. To evaluate the performance of this proposed one with the other counterparts (DES, DESX, 2DES, 3DES and AES), we taken the measures like Time for Encryption/Decryption, Simple regression, Correlation analysis, Hamming distance and Disparity of cipher.

Time for Encryption/Decryption: It is one of the important considerations while designing an algorithm. Even though an algorithm is more and more strength, if it takes much encryption time, it does not become a Best choice. The Table 1 which is below depicts the time required for the encryption process for DES, DESX, 2DES, 3DES and AES.

Table 1: Time taken for Encryption/Decryption

S.No	Encryption Algorithms	Time taken (in Sec)
1	DES	1
2	DESX	6
3	2 DES	6
4	3 DES	8
5	AES	9
6	EDES-H	4

As the below figure illustrates, AES takes much Highest time to take for Encryption/Decryption because of its key length is high. It uses the key size of 128,192,256 bits. Even though it provides more security when it uses the key size of 256 bits, it takes much time and an average taking time of AES is also much high when compared to all other encryption algorithms. Coming to 3DES, it also takes much time because of it uses 3 keys to encrypt and decrypt, which is a little bit less time taken when compared to AES. The 2 DES and DESX would take an approximate of the same time because the 2 DES uses 2 keys to encrypt and decrypt and DESX augments DES by XOR ing an extra 64 bits of key (K_1) to the plaintext *before* applying DES, and then XOR ing another 64 bits of key (K_2) *after* the encryption: The key size is thereby increased to 56 + (2 × 64) = 184 bits. When compared to all other encryption algorithms, HDES takes a little bit less time when compared to all DES Variants and a bit delay when compared to

ISSN 2348 – 8034
Impact Factor- 4.022

the DES because, it's the hash component and extra processing because of the adding the secure bits for the S-boxes and modification of the Expansion function. The below shown the Fig 3. Clearly illustrates the performance Comparison for the Encryption/Decryption of the EDES with its counterparts.

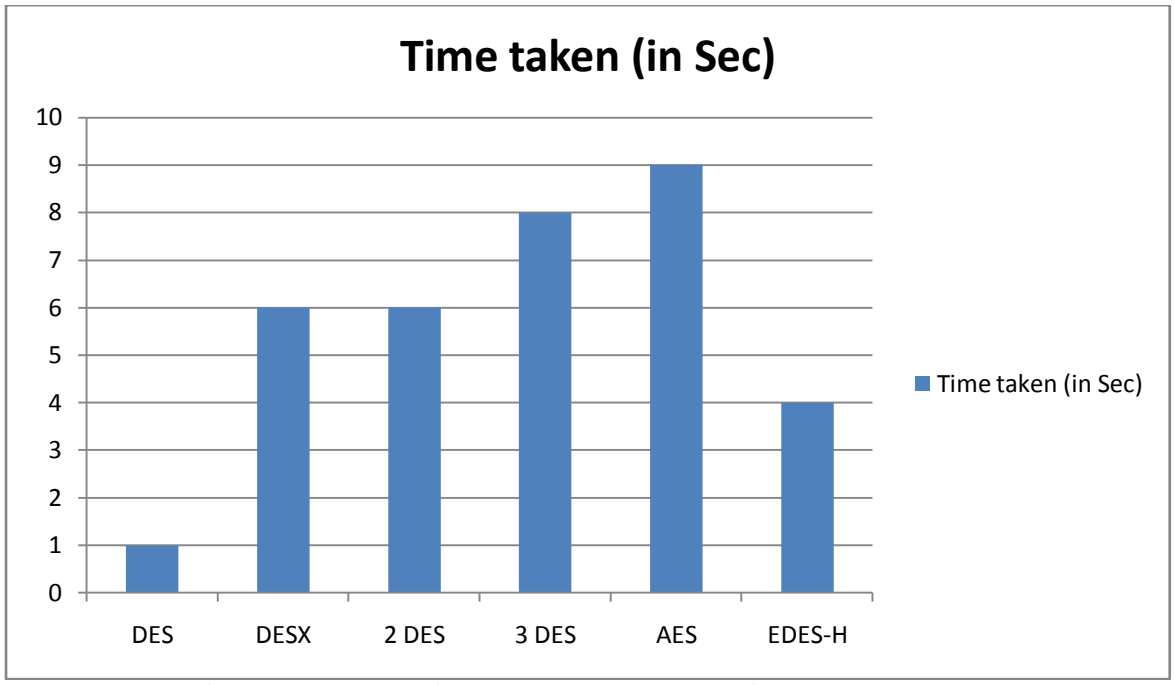

Fig 3: performance Comparison for the Encryption/Decryption of the EDES with its counterparts.

Simple regression & Correlation analysis Tests: It is a common statistical data analysis technique used to tests how the change in the predictor variable predicts the level of change in the outcome variable. This test is used to conclude the point to which there is a relationship between a dependent variable and an independent variable. The difference between the two is the number of independent variables.

In simple regression, there are two variables P and Q, wherein Q depends on P or it may be influenced by P. Here Q is called as dependent, or decisive factor variable and P is independent or predictor variable. The regression line of Q on P can be expressed as: **Q = Ka + Bp**

Where K = constant, b = regression coefficient,

In this equation, a and b are the two regression parameters.

Correlation analysis Test is a method of statistical evaluation used to study the strength of a relationship between two variables. It is used to compare an experimental data with the data what we are expecting from the system based on the Pre-defined assumption. This type of analysis is mostly useful when a researcher wish to set up if there are possible links between variables. This assumption generally commences before starting the experiment depending on the assessors indulgent and viewpoint regarding the assumptive numerical conclusion of the particular research.

The correlation analysis test is normally functional to check whether there is any difference between two or more groups of data.

The formula for the correlation coefficient is

$$r = \frac{\mathrm{Cov}(x,y)}{\sqrt{s_x^2 * s_y^2}}$$

[Ganesh, 5(7): July 2018] ISSN 2348 – 8034
DOI: 10.5281/zenodo.1310796 Impact Factor- 4.022

Where Cov(x,y) is the covariance of x and y defined as

$$\mathrm{Cov}(x, y) = \frac{\Sigma(X - \overline{X})(Y - \overline{Y})}{n - 1}$$

s_x^2 and s_y^2 are the sample variances of x and y, defined as

$$s_x^2 = \frac{\Sigma(X - \overline{X})^2}{n - 1} \quad \text{and} \quad s_y^2 = \frac{\Sigma(Y - \overline{Y})^2}{n - 1}$$

Table 2: Feasible Values for the Simple regression Analysis and Correlation analysis for the encryption algorithms

Encryption Algorithm	Simple Regression Test %	Correlation analysis Test %
DES	97.5	97.4
DESX	96.8	96.6
2 DES	96.8	96.9
3 DES	95.4	95.5
AES	94.6	94.8
EDES-H	89.1	89.1

In the cipher data is anticipated to have half of zeroes and half of ones, because in an actual random function the ratio between zeros and ones are almost equal. The cipher data is expected to have 50% of zeroes and 50% of ones, because in the random function the ratio between zeros and ones must be equal. Table 2 illustrates the Simple regression analysis test prediction based on their related values with degree of choice. If the feasible value is greater than 98% or less than 2%, then the cipher text is almost not random. If the feasible values are between 98% and 90% the cipher text can be assumed to be reasonably random. If the feasible values between 90% and 80% specify the cipher text is mostly random. The Table 2 shows the feasible values of the DES and its Variants.

Based on the Illustration of the above Table 2, EDES-H has the surpassed level of randomness when compared to the DES and its variants with a feasible value of 89.1% consistently with both the Simple regression Test and correlation analysis Tests. This clearly indicates the cipher text obtained from HDES is mostly random. Where the counterparts of the EDES-H are DES, DESX, 2DES, 3DES and AES fall under the category of 90% to 98% which are assumed to be reasonably random and the feasible values of these algorithms are varied a little bit with the simple regression and correlation analysis tests. The performance comparison is depicted in the Fig 4.

[Ganesh, 5(7): July 2018]
DOI: 10.5281/zenodo.1310796
ISSN 2348 – 8034
Impact Factor- 4.022

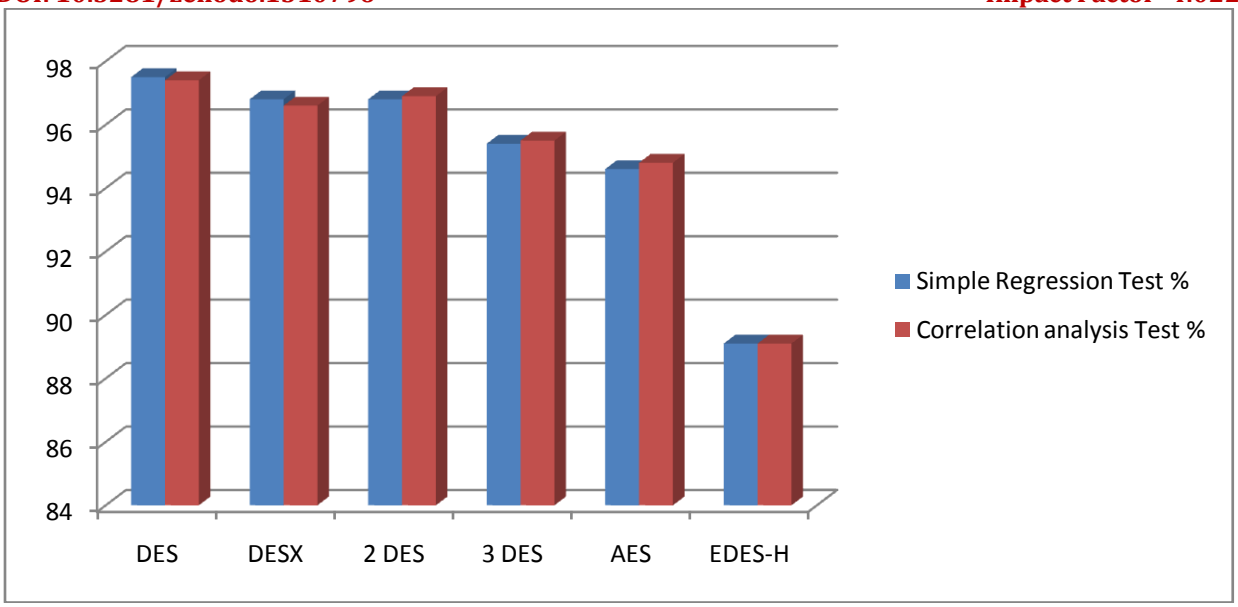

Fig 4: The performance comparison of EDES-H for the Feasible Values of the Simple regression Analysis and Correlation analysis Tests with the various encryption algorithms

Hamming Distance: The hamming distance is one of the critical exercises to incarcerate the diffusion effect for the encryption algorithms. The Complicated diffusion is a best measure for any algorithm, which changes the plaintext bit or key bit changes a big change in cipher text. It is calculated using the below Equation,

$$\text{Hamming distance } (P, Q) = |\{l|\ P_i \neq Q_j\}|$$

Where P is the first cipher text which consists of several bits say P1, P2… Pn, Q is the second cipher text which consists of several bits say Q1, Q2… Qn. The hamming distance is calculated by flipping a number of bits in the plaintext:

$$\text{Hamming distance } (P, Q) = |\{l|\ P_i \neq Q_j\}|$$

The hamming distance is computed between the generated cipher texts corresponding to the plain text with flipped bits i.e., as 2 bits, 4 bits and 8 bits. This process is performed on each of the algorithms DES, DESX, 2DES, 3DES and AES. As the Practical observation reveals that the EDES-H has a slight higher hamming distance than its counterparts in almost all the cases.

Disparity of Cipher: It is another principal measure to identify the randomness of the data encryption algorithms. It calculates by detecting the absolute disparity of ones and zeros in the plaintext and its corresponding disparity in the cipher text. If we get the less value with this measure this metric denotes more randomness. Fig. 5 clearly depicts the disparity of cipher DES, DESX, 2DES, 3DES and AES. Various entropies of data are used to assess the disparity of cipher of the different encryption algorithms.

The Fig. 5 presents by taking an average of 25 runs for each and every encryption algorithm. As the figure shows, EDES-H demonstrates lower cipher difference of ones and zeros in most of the cases against to all other encryption algorithms. This clearly indicates that the randomness obtained from EDES-H outperforms their counterparts.

ISSN 2348 – 8034
Impact Factor- 4.022

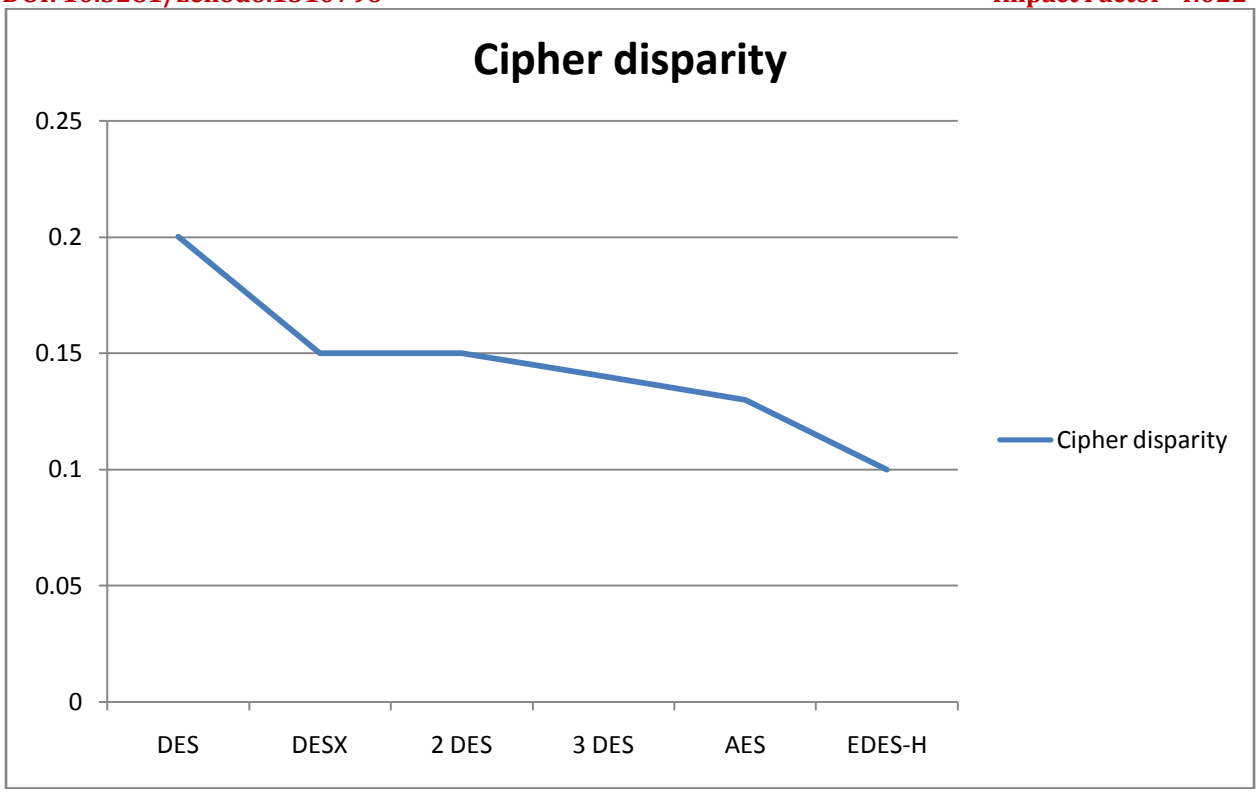

Fig 5: Comparison of the Disparity of Cipher of EDES-H with various Encryption algorithms

V. CONCLUSION AND FUTURE SCOPE

It is necessary to provide a secure communication for data transmission. DES is now considered to be an insecure technique of encryption for most of the applications because some analytical results demonstrates the theoretical weaknesses in the cipher. So it needs very important to enhance this algorithm by adding a new level of security by modifying the Expansion functions and increasing the randomness for S-Box operation which increases the confusion parameter of the algorithm. Hence, it reflects in the security of the algorithm. In future, we can implement this algorithm by modifying the other functions like S-box designing by replacing XOR operation with some other transposition etc., the modification of Expansion function, Permutation Function and etc.,

This paper proposes by modifying the DES and named as EDES-H by adding a Hashing component to the DES and enhancing the randomness for the S-Box and modifying the Expansion function. The objective of EDES-H is to overcome the imperfection in the original DES, by redesigning and combining several techniques and components to enhance the original DES. The main characteristic of EDES-H is its hash component to generate a unique finger mark and enhancing the internal components such as Expansion function along with increasing randomness to make the S-Boxes more secure for each round of the encryption process. The EDES-H evaluated extensively using a number of metrics: Time for Encryption/Decryption, Simple regression, Correlation analysis, Hamming distance and Disparity of cipher and the performance has been compared against DES, DESX, 2 DES, 3DES and AES. This have been confirmed by the conducted experiments that show HDES with higher degree of randomness in terms of Time for Encryption/Decryption, Simple regression, Correlation analysis, Hamming distance and Disparity of cipher. In future work, it is better to explore the other randomness measures like linear Complexity test, discrete Fourier transform (spectral) test and Fast Fourier transform test approximate entropy test of EDES-H against IDEA, CAST and blowfish algorithms.

[Ganesh, 5(7): July 2018] ISSN 2348 – 8034
DOI: 10.5281/zenodo.1310796 Impact Factor- 4.022

REFERENCES

1. *W. Stallings, "Cryptography and Network Security: Principles and Practices", 5th ed., Prentice Hall, 1999*
2. *Schneier, B., "Applied Cryptography: Protocols, Algorithms and Source Code in C". 2nd Edn., John Wiley & Sons, Inc., Indianapolis, IN, ISBN-10: 0471128457, 1996, pp: 758.*
3. *Matsui, M., "On Correlation between the Order of S-Boxes and the Strength of DES". In: Advances in Cryptology—EUROCRYPT'94, De Santis, A. (Ed.), Springer, ISBN-10: 978-3-540-60176-0, 1994, pp: 366-375.*
4. *Biham, E. and A. Biryukov. "How to strengthen DES using existing hardware" In: Advances in Cryptology ASIACRYPT'94, Pieprzyk, J. and R. Safavi-Naini (Eds.), Springer, Berlin, Heidelberg, ISBN-10: 978-3-540-59339-3, 1995, pp: 398-412.*
5. *Zhuang, Z., J. Chen and H. Zhang, "A countermeasure for DES with both rotating masks and secured S-boxes". Proceedings of the 10th International Conference on Computational Intelligence and Security, Nov. 15-16, 2014. IEEE Computer Society, Washington, pp: 410-414. DOI: 10.1109/CIS.2014.43*
6. *Al-Qassas, R.S., M. Qasaimeh and H. Al-Nouri. 2016. "A fingerprint featured data encryption algorithm". Proceedings of the 7th International Conference on Information and Communication Systems, Apr. 5-7, IEEE Xplore Press, Jordan, pp: 227-232. DOI: 10.1109/IACS.2016.7476116*
7. *Malik Qasaimeh, Raad S. Al-Qassas 2017, "Randomness Analysis of DES Ciphers Produced with Various Dynamic arrangements", Journal of Computer Science. DOI: 10.3844/jcssp.2017.*
8. *Dr. Mohammed M. Alani, "Improved DES Security", International Multi-Conference on System, Signals and Devices, 2010.*
9. *Knopf, C., 2007, "Cryptographic Hash Functions", Leibniz University, Hannover, Germany*