HYBRID DATA ENCRYPTION STANDARD

*1 Ms. Priya S, *2 Ms. Anita Madona M

*1 M.Phil Research Scholar, Department of Computer Science Auxilium College (Autonomous), Vellore, TamilNadu, India

*2 Assisant Professor, Department of Computer Science Auxilium College (Autonomous), Vellore, TamilNadu, India

---

Abstract - Security is playing an important role in the field of network communication system. Cryptography is the branch of computer science that deals with hiding information for secure communication of data. It uses the codes to convert plain text into cipher text, so that only the intended recipient will be able to read it using the key. The cryptographic algorithms are mainly divided into two categories as Symmetric and Asymmetric on the basis of using the same or different key for encryption and decryption. The most popular and widely used symmetric-key system is the Data Encryption Standard (DES) in *which both the sender and receiver use a shared secret key to encrypt or decrypt the data.*

DES is the block cipher which takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. The key basically consists of 64 bits however, only 56-bits of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56-bits. DES is now considered to be insecure for many applications. This is due to the 56-bit key size being too small which is not so powerful against the Brute force attack.

To improve the security of DES algorithm and to avoid the Brute force attack, the substitution technique- Affine Cipher is used to mask the plaintext and then pass it as input to the DES algorithm to perform encryption. This Affine Cipher is used before the original DES algorithm, to make cryptanalysis difficult and improve the security of DES. The Simulation tool NS-2 is used to compare and analyze the performance of DES and the Hybrid DES.

Keyword: DES – Data Encryption Standard, NS – Network Simulator, HDES – Hybrid Data Encryption Standard

1. INTRODUCTION

The most widely used Symmetry key cryptographic technique is DES. DES is the block cipher which takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. It is a symmetric encryption technique which means both sender and receiver can use a shared key to encrypt and/or decrypt the data. The key basically consists of 64 bits however, only 56-bits of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56-bits.

Nowadays, the parallel processor and advanced computer machines are discovered which can perform the computation and calculation at very high speed. Though DES is a very powerful algorithm, these machines can break DES. DES is now considered to be insecure for many applications. This is due to the 56-bit key size being too small and is not so powerful against the Brute force attack. A brute force attack against a cipher consists of breaking a cipher by trying all possible keys. Statistically, if the keys were originally chosen randomly, the plaintext will become available after about half of the possible keys are tried. For any cipher, the most basic method of attack is brute force i.e. trying every possible key in turn.

Hence to improve the security of DES algorithm, the substitution technique- Affine Cipher is used to mask the plaintext and then pass it as input to the DES algorithm. Thus the input text is scrambled by means of Affine Cipher and made secured before given as input to the DES, to make cryptanalysis difficult. If this technique is used before the original DES algorithm then the intruder required to break the DES algorithm first and then the Affine Cipher. Thus Brute Force attack is made complicated. This method improves security in DES. So the security is approximately double as compared to a DES algorithm.

2. HYBRID DATA ENCRYPTION STANDARD

DES seems to be weak against the brute force attacks. To improve the security of DES algorithm, the substitution technique - Affine Cipher is added before the DES algorithm to mask the input to DES in order to improve the security of DES. The Affine Cipher is used to create the cipher text which is used as input text to the DES. The plaintext which is going to be give as input to the DES is masked by means of Affine Cipher.

Thus the input text of the DES itself is cipher text, which is going to be act as a safe input to DES. Hence the

security of the DES seems to be double when comparing to the DES algorithm.

2.1 Encryption Process of HDES

In HDES the plain text is given as input to Affine Cipher. The Affine Cipher is used to create the cipher text. The first step in the encryption process of Affine is to transform each of the letters in the plaintext alphabet to the corresponding integer in the range 0 to *m*-1, (i.e.) the encryption process for each letter is given by E(*x*) = (*ax* + *b*) mod *m*, where *a* and *b* are the key for the cipher text.

This means that we multiply the integer value for the plaintext letter by *a*, and then add *b* to the result. Finally, we take this modulus *m* (i.e.) the remainder is taken as a solution to divided by *m*, or the length of the alphabet is taken until the number less than its length. The resultant masked/cipher text will be given as input to the DES. In DES, the plain text will be given as input but in HDES, the input text of the DES itself is cipher text. After giving the input to the DES, the DES performs its normal working function.

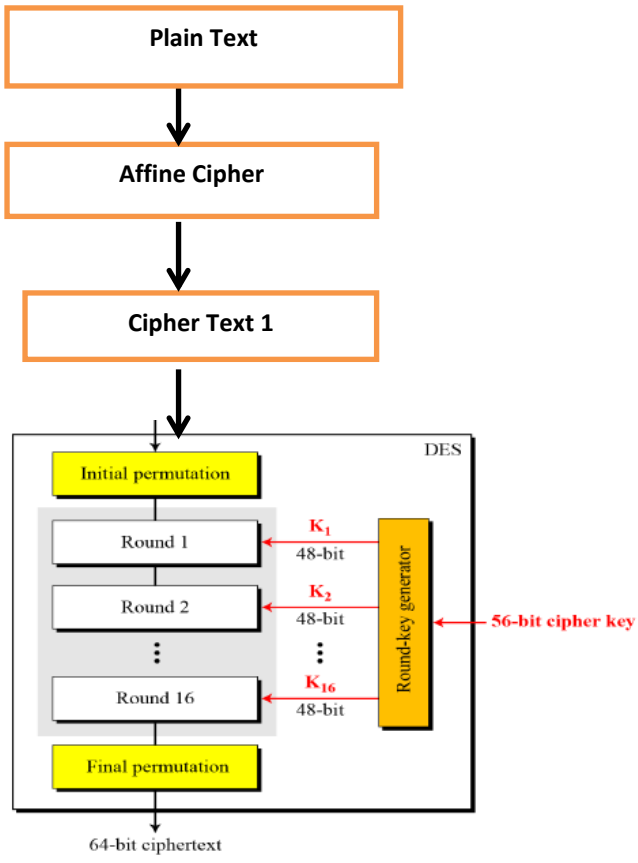

Figure 2.1 Encryption Process of HDES

2.2 Decryption Process of HDES

Decryption uses the same algorithm as encryption, except that the sub keys K1, K2 ...K16 are applied in reversed order. For decryption, the cipher text will be given as the inputs to the DES algorithm but use the keys K_i in reverse order. That is, K_{16} on the first iteration, K_{15} on the second until k_1 which is used on the 16th and last iteration.

After the last decryption process of the DES, the cipher text will be obtained. This cipher text should be entered into Affine Cipher for further decryption process in Affine. In deciphering the cipher text in Affine, the opposite (or inverse) functions has to be applied on the cipher text to retrieve the plaintext. The first step is to convert each of the cipher text letters into their integer values.

We must now perform the following calculation on each integer D(*x*) = *c*(*x* - *b*) mod *m*, where *c* is the modular multiplicative inverse of *a*. That is, *a* x *c* = 1 mod *m* (*c* is the number such that when you multiply a by it, and keep taking away the length of the alphabet, you get to 1. Then the original plain text is got from the reverse of HDES algorithm.

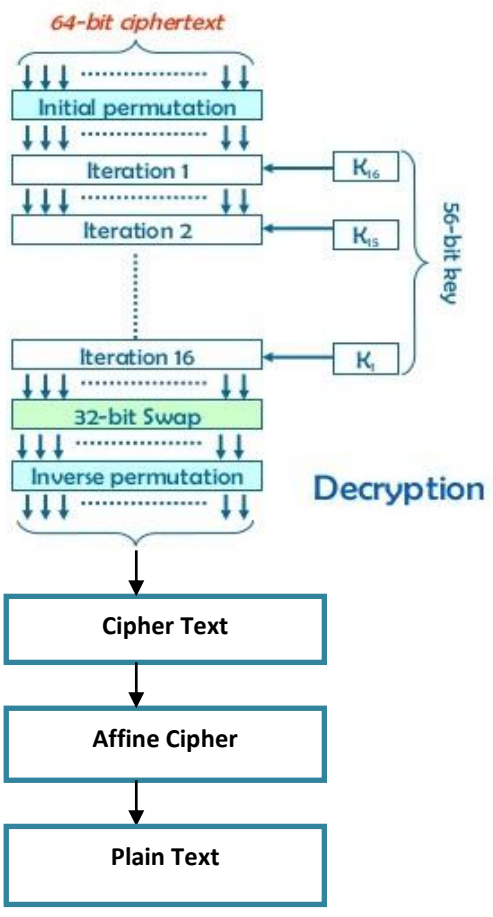

Figure 2.2 Decryption Process of HDES

3. HDES ALGORITHM

Step 1: Get plain text X

Step 2: While X not equal to null, repeat step 3 to step 6

Step 3: Set A=0 to Z=25 and n=25

Step 4: Get key1 a and key2 b

Step 5: Operate C = (ax+b) mod n

Step 6: Write cipher text C

Step 7: The cipher text C is given as input text to the Initial Permutation (IP) function of DES.

Step 8: The Initial permutation is performed on C.

Step 9: The initial permutation produce two halves of permuted block: Left Plain Text (LPT) and Right Plain Text (RPT).

Step 10: Each of LPT and RPT goes through 16 rounds of encryption process, with its own key:

10.1: From the 56-bit key, a different 48-bit Sub-key is generated using Key Transformation.

10.2: Using the Expansion Permutation, the RPT is expanded from 32 bits to 48 bits.

10.3: The 48-bit key is XORed with 48-bit RPT and resulting output is given to the next step.

10.4: The S-box substitution produces 32-bit output from 48-bit.

10.5: The P-Box Permutes these 32 bits.

10.6: The P-Box output 32 bits are XORed with the LPT 32 bits.

10.7: The result of the XORed 32 bits are become the RPT and old RPT become the LPT. This process is called as Swapping.

10.8: Then LPT and RPT is given to the next round and continues to perform the 15 rounds.

Step 11: After the completion of 16 rounds the Final Permutation is performed, which to give the 64-bit secured cipher text.

In the above algorithm, the Affine Cipher is added to mask the plaintext before given as input to the DES algorithm to improve the security of DES. Hence the security of the HDES seems to be double when compared to existing DES algorithm.

4. SIMULATION RESULT

The original Data Encryption Standard algorithm and the HDES are implemented using NS2 simulator. The experimental results are shown below:

Encryption Time

Table 4.1 shows the encryption time in milliseconds against 20 milliseconds of simulation time for DES and HDES.

Algorithm	Encryption Time
DES	214 milliseconds
HDES	262 milliseconds

Table 4. 1 Encryption Time

Figure 4.1 shows the graph encryption time comparison for the two different algorithms against 10-50 milliseconds of simulation time where x-axis shows the simulation time (sec) and y-axis shows the encryption time (ms).

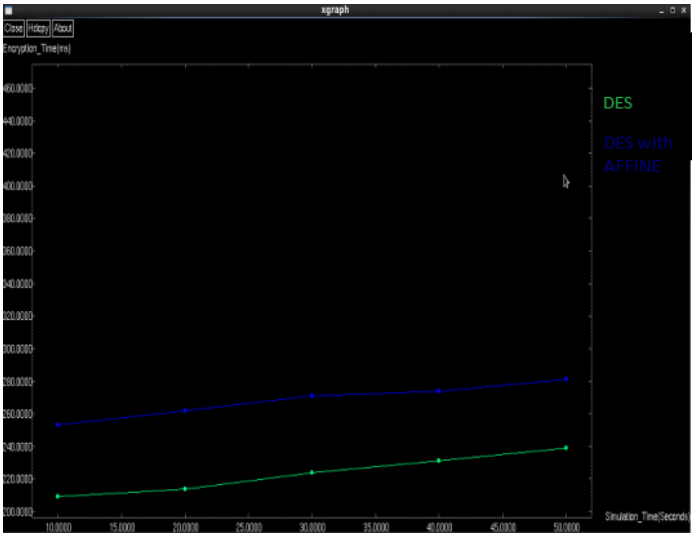

Figure 4.1 Encryption Time

From the Table 4.1 and Figure 4.1, it is observed that the encryption time of HDES is more when compared to DES, because Affine Cipher and DES are combined together as one algorithm HDES.

Decryption Time

Table 4.2 shows the decryption time in milliseconds against 20 milliseconds of simulation time for DES and the combined algorithm of Affine Cipher and DES.

Algorithm	Decryption Time
DES	221 milliseconds
HDES	253 milliseconds

Table 4.2 Decryption Time

Fig 4.2 shows the graph for comparison of decryption time for the two different algorithms against 10-50 milliseconds of simulation time where x-axis shows the simulation time (sec) and y-axis shows the decryption time (ms).

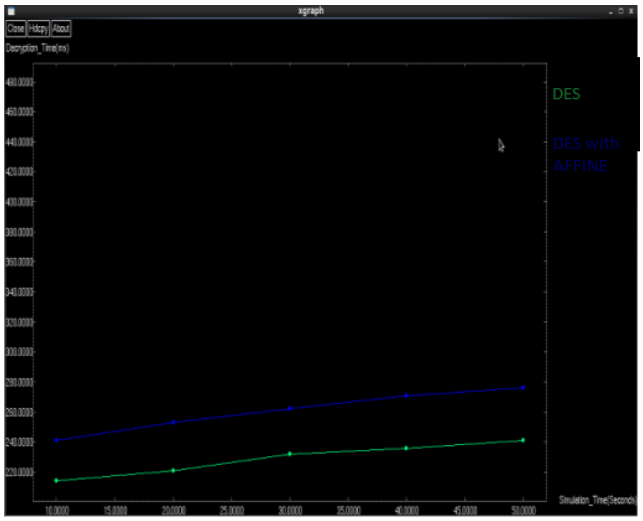

Figure 4. 2 Decryption Time (MS)

From the Table 3.2 and Figure 3.2, it is observed that the decryption time of HDES is more when compared to DES, because Affine Cipher and DES are combined together as one algorithm HDES

Throughput

Table 4.3 shows the throughput in kbps against 20 milliseconds of simulation time for DES and the combined algorithm of Affine Cipher and DES.

Algorithm	Throughput
DES	28.13 milliseconds
HDES	35.20 milliseconds

Table 4.3 Throughput

Fig. 4.3 shows the graph for comparison of throughput for the two different algorithms against 10-50 milliseconds of simulation time for various algorithms where x-axis shows the simulation time (sec) and y-axis shows the throughput (kbps).

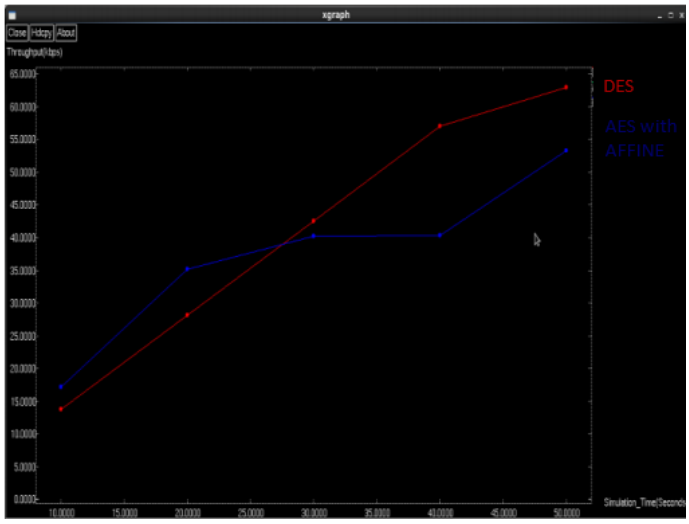

Figure 4.3 Throughput (kbps)

From the Table 4.3 and Figure 4.3, it is observed that the throughput time of HDES is more when compared to DES, because Affine Cipher and DES are combined together as one algorithm HDES

Transmission Time

Table 4.4 shows end to end delay against 20 milliseconds of simulation time for DES and the combined algorithm of Affine Cipher and DES.

Algorithm	Transmission Time
DES	48.7766 milliseconds
HDES	48.8349 milliseconds

Table 4.4 Transmission Time

From the Table 4.4, it is observed that transmission time of HDES is more when compared to DES, because Affine Cipher and DES are combined together as one algorithm HDES.

Timing complexity of breaking DES and DES with AFFINE:

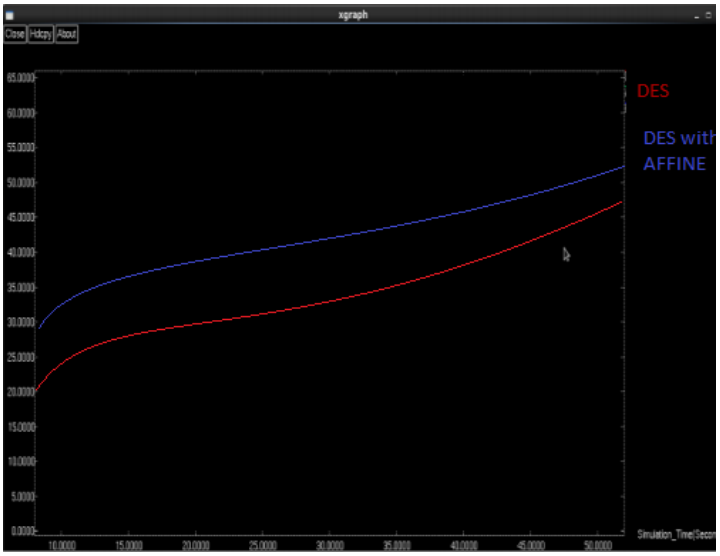

Figure 4.4 Time need to break cipher

From the graph we can observe that the computation time taken by the HDES is more than DES. Though the extra computation time is needed by the HDES, the security level is higher than the DES algorithm, (i.e.) it takes more time to break the cipher text of HDES.

CONCLUSION

The original DES implementation has some weaknesses. To overcome this weakness the Affine Cipher is added in DES algorithm. The Designed system improves the security of original DES. The only drawback is increase in computation time, this can also be avoided by having parallel and high speed computation power. By using Affine Cipher in DES, the security of DES algorithm is very tight and approximately impossible to break the DES algorithm.

In the earlier approaches when Affine Cipher and DES algorithm used separately there were chances for the cryptanalysis to break the code by the third party. After analyzing both of these techniques we came to the conclusion that neither of the technique is much secure. But a combination of both of these techniques can provide much better security.

The simulation result shows that, it is much difficult to break the cipher in Dual Security of DES when comparing to simple DES. Because the time taken to break the HDES is more complex than simple DES. When both of them are used in one approach individually they tend to be more powerful and economical.

ACKNOWLEDGEMENT

I thank the Almighty God Yehova who showered his immense blessings, which helped me to complete this dissertation successfully. I am indebted to Ms. Anita Madona M., Asst. Prof. Department of Computer Science, Auxilium College (Autonomous), Vellore, who guided my dissertation with reviews, evaluations, guidance and suggestions throughout this dissertation. She offered her time to perfect my work. I appreciate her immense patience. I express my Whole hearted thanks to my Parents and Siblings for their encouragement to bring this dissertation to a successful completion.

REFERENCES

[1]Ruah Praise .Y, Shishir Shukla, *"Implementation of Affine Substitution Cipher with Keyed Transposition Cipher for Enhancing Data Security",* IEEE, Special issued on the protection of data, vol. 4, Jan 2014.

[2] M.E. Hellman, *"DES will be Totally Insecure within Ten Years"*, IEEE Spectrum, Vo1.16, N0.7, pp32 -39, July 1979.

[3] Alani, M.M., *"A DES96 - Improved DES Security"*, 7th International Multi-Conference on Systems, Signals and Devices, Amman, 27-30 June 2010.

[4]Adam. J, *"Threats and Countermeasures"*, IEEE Spectrum, vol. 29, August 1992, 21-28.

[5]De Millo. R, Merritt. M, *"Protocols for Data Security"*, IEEE Computer, vol.16, February 1983, 39 -54.

[6] Evans. A, *"Comparing Information without Leaking IT"*, Common. Of the ACM, Vol.39, May 1996, 77-85.

[7]Hellman. M, *"An Overview of Public Key Cryptography"*, IEEE communication society Magazine, vol. 16, November 1978, 24-32.

