International Journal of Management, IT & Engineering
(ISSN: 2249-0558)

CONTENTS

Chief Patron

Patron

Editorial Board

Editorial Advisory Board

Mr. P. PRABHU
Assistant Professor, Alagappa University, KARAIKUDI

Mr. MANISH KUMAR
Assistant Professor, DBIT, Deptt. Of MBA, DEHRADUN

Mrs. BABITA VERMA
Assistant Professor, Bhilai Institute Of Technology, DURG

Ms. MONIKA BHATNAGAR
Assistant Professor, Technocrat Institute of Technology, BHOPAL

Ms. SUPRIYA RAHEJA
Assistant Professor, CSE Department of ITM University, GURGAON

September 2011

IJMIE

Volume 1, Issue 4

ISSN: 2249-0558

Title

AN EFFICIENT IMPLEMENTATION OF TRIPLE DES (DATA ENCRYPTION STANDARD) THROUGH HASH FUNCTION

Author(s)

N. Venkatesan

Dept. of IT,

Bharathiyar college of Engg. and Technology,

Karaikal

Abstract:

Cryptography, being one of the techniques that is handed out for securing a network. It is utilized by implementing many different algorithms. DES (Data Encryption Standard) is one of the most popular algorithms. But a network implementing DES can be attacked. Hence Triple DES and AES (Advanced Encryption Standard) were formulated to overcome some of the shortcomings of DES. Even then loop holes were found in Triple DES too. In order to plug these loop holes, the new approach Triple DES in a secured way by injecting message digest to it. This paper uses the conventional EDE (Encryption Decryption Encryption) technique as used in triple DES. With the help of two keys transmission, network itself processes with three keys each of 56 bits in length internally and also describes the effectiveness of new approach.

Keywords: Data Encryption Standard, Triple Data Encryption Standard, Message Digest

Introduction:

Cryptography is an art and science of keeping messages secure. When a message is transferred from one place to another, its contents are readily available to an eavesdropper. A simple network-monitoring tool can expose the entire message sent from one computer to another in a graphical way. For an N-Tier or distributed application to be secure, all messages sent on the network should be scrambled in a way that it is computationally impossible for any one to read it.

In secret key cryptography, a single key is used for both encryption and decryption. The major difficulty of secret-key algorithm is the need for sharing the secret-key. Asymmetric key encryption uses different keys for encryption and decryption. The encryption algorithm performs various substitutions and transformations on the plaintext i.e. the original intelligible message or data that is fed into the algorithm as input. Use of secret key, input text is converted into cipher text. The decryption algorithms run in reverse. It takes the cipher text and the secret key in order to produce the original plain text.

Using DES cryptosystem for encryption and decryption purpose yields some disadvantages. To overcome this, Triple DES is introduced. Even though, Eavesdroppers

scrambled the messages. Instead of using AES (Advanced Encryption Standard) for secured transmission, an idea is given to Triple DES by hybridizing hash function. So, user need not switch over to others. This paper describes how to merge hash function and secret key crypto system algorithm Triple DES.

Section 2 describes the overview of cryptography. Section 3 deals with the basic concepts of Triple DES cryptosystem and hash functions. Section 4 discusses need the new idea of how to merge two types of cryptosystems and the resulting new approach of Triple DES. Section 5 deals with performance analysis of new Triple DES. The study is concluded in the section 6, along with the future work.

Overview of Cryptography:

There are several ways of classifying cryptographic algorithms.

- **Secret Key Cryptography**: Uses a single key for both encryption and decryption
- **Public Key Cryptography**: Uses one key for encryption and another key for decryption
- **Hash Functions**: Uses a mathematical transformation to irreversibly "encrypt" information

In secret key cryptography [2], a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the cipher text to the receiver. The receiver applies the same key (or ruleset) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption.

With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key. The algorithm used for symmetric key encryption is called secret-key algorithm. Since secret-key algorithms are mostly used for encrypting the content of the message. They are also called content-encryption algorithms. Strength of the symmetric key encryption depends on the size of the key used. For the same algorithm, encrypting using longer

key is tougher to break than the one done using smaller key. Strength of the key is not liner with the length of the key but doubles with each additional bit.

There are two types of secret-key ciphers, i.e., block ciphers and stream ciphers. Block Ciphers convert fixed-length block of plain text into cipher text of the same length. Stream Ciphers operate on small group of bits, typically applying bitwise XOR operations to them using the key as a sequence of bits. Following are some block ciphers with their normal block size.

DES - 64 bits 3DES – 64 bits AES – 128 bits

Basic Concepts:

For any Secret Key Cryptosystem transmission, 2 inputs are used. One is Plaintext and another one is key [6]. Key length is varied depending upon usage of algorithms. For any hash function, it is unbreakable mathematical function applied for only one input that is plaintext. Hash function used for any key length and produce an output as cipher text.

Triple DES:

As early as 1979, IBM realized that the DES key length was too short and devised a way to effectively increase it, using triple Data Encryption Standard [2, 3]. The method chosen, which has since been incorporated in International Standard 8732, is illustrated in Figure 3.1. Here two keys and three stages are used. In the first stage, the plain text is encrypted by using DES in the usual way with K1. In the second stage, DES is run in decryption mode, using K2 as the key. Finally, another DES encryption is done with K1.

This design immediately gives rise to two questions. First, why are only two keys used, instead of three? Second, why is EDE (Encrypt Decrypt Encrypt) used, instead of EEE (Encrypt Encrypt Encrypt)? The reason that two keys are used is that even the most paranoid cryptographers believe that 112 bits is adequate for routine commercial applications for the time being. (And among cryptographers, paranoia is considered a feature, not a bug.) Going to 168 bits would just add the unnecessary overhead of managing and transporting another key for little real again.

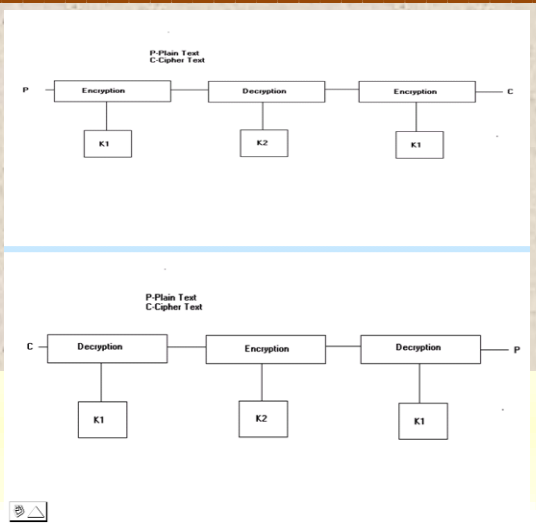


Figure 3.1: Triple Data Encryption Standard

Therefore, it is reasonable to assume that if DES is used twice with different keys, it will produce one of the many mappings that is not defined by a single application of DES. The above diagram represents two keys with three stages encryption and decryption process. This is only alternative to DES and has adopted for use in the key management standards. There are no practical cryptanalytic on triple DES. Existing DES procedure is followed.

DES Encryption:

The overall scheme for DES encryption is shown below. There are two inputs to the encryption function, the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

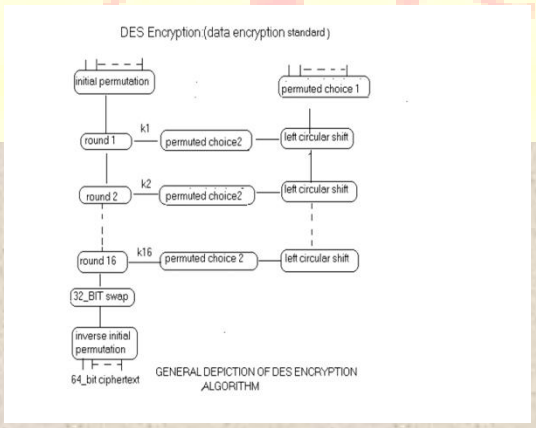


Figure 3.2: DES algorithm description

Looking at the left-hand side of figure 3.2, we can see that the processing of the plaintext proceeds in three phases. First, the 64 –bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input. This is followed by a phase consisting of 16 round of same function, which involves both permutation and substation function. The output of the last round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the pre output. Finally, the pre output is passed through a permutation (IP^{-1}) that is the inverse of the initial permutation function; to produce the 64- bit cipher text with the exception of the initial and final permutation DES has the exact structure of the Feistel cipher.

IP: Initial Permutation

Bit	0	1	2	3	4	5	6
1	58	50	42	34	26	18	10
9	60	52	44	36	28	20	12
17	62	54	46	38	30	22	14
25	64	56	48	40	32	24	16
33	57	49	41	33	25	17	9
41	59	51	43	35	27	19	11
49	61	53	45	37	29	21	13
57	63	55	47	39	31	23	15

IP^(-1): Inverse Initial Permutation

Bit	0	1	2	3	4	5	6	7
1	40	8	48	16	56	24	64	32
9	39	7	47	15	55	23	63	31
17	38	6	46	14	54	22	62	30
25	37	5	45	13	53	21	61	29
33	36	4	44	12	52	20	60	28
41	35	3	43	11	51	19	59	27
49	34	2	42	10	50	18	58	26
57	33	1	41	9	49	17	57	25

The right-hand portion of figure 3.2 shows the way in which the 56 –bit key is used. Initially, the key is passed through a permutation function. Then, for each of the 16 rounds, a sub key (k_i) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different sub key is produced because of the repeated iteration of the key bits.

Initial Permutation:

The input to a table consists of 64 bits numbered from 1 to 64. The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64. Each entry in the permutation table indicates the position of a numbered input bit in the output, which also consists of 64 bits.

PC-1: Permuted Choice 1

Bit	0	1	2	3	4	5	6
1	57	49	41	33	25	17	9
8	1	58	50	42	34	26	18
15	10	2	59	51	43	35	27
22	19	11	3	60	52	44	36
29	63	55	47	39	31	23	15
36	7	62	54	46	38	30	22
43	14	6	61	53	45	37	29
50	21	13	5	28	20	12	4

PC-2: Permuted Choice 2

Bit	0	1	2	3	4	5
1	14	17	11	24	1	5
7	3	28	15	6	21	10
13	23	19	12	4	26	8
19	16	7	27	20	13	2
25	41	52	31	37	47	55
31	30	40	51	45	33	48

37	44	49	39	56	34	53
33	46	42	50	36	29	32

Details of Single Round:

Key Scheduling:

Although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. The least significant (right-most) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits.

E-Bit Selection Table

Bit	**0**	**1**	**2**	**3**	**4**	**5**
1	32	1	2	3	4	5
7	4	5	6	7	8	9
13	8	9	10	11	12	13
19	12	13	14	15	16	17
25	16	17	18	19	20	21
31	20	21	22	23	24	25
37	24	25	26	27	28	29
43	28	29	30	31	32	1

P Permutation

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

The first step is to pass the 64-bit key through a permutation called Permuted Choice 1, or PC-1 for short. The table for this is given below. Note that in all subsequent descriptions of bit numbers, 1 is the left-most bit in the number, and n is the rightmost bit.

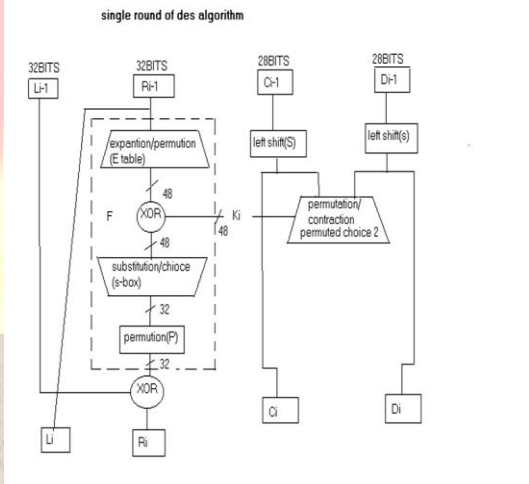


Figure 3.3: Single round DES algorithm

For example, we can use the PC-1 table to figure out how bit 30 of the original 64-bit key transforms to a bit in the new 56-bit key. Find the number 30 in the table, and notice that it

belongs to the column labeled 5 and the row labeled 36. Add up the value of the row and column to find the new position of the bit within the key. For bit 30, 36 + 5 = 41, so bit 30 becomes bit 41 of the new 56-bit key. Note that bits 8, 16, 24, 32, 40, 48, 56 and 64 of the original key are not in the table. These are the unused parity bits that are discarded when the final 56-bit key is created. Now that we have the 56-bit key, the next step is to use this key to generate 16 48-bit sub keys, called K[1]-K[16], which are used in the 16 rounds of DES for encryption and decryption. The procedure for generating the sub keys - known as key scheduling - is fairly simple:

Subkey Rotation Table

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of bits to rotate	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES Core Function:

Once the key scheduling and plaintext preparation have been completed, the actual encryption or decryption is performed by the main DES algorithm. The 64-bit block of input data is first split into two halves, L and R. L is the left-most 32 bits, and R is the right-most 32 bits. The following process is repeated 16 times, making up the 16 rounds of standard DES. We call the 16 sets of halves L[0]-L[15] and R[0]-R[15].

1. R[I-1] - where I is the round number, starting at 1 - is taken and fed into the E-Bit Selection Table, which is like a permutation, except that some of the bits are used more than once. This expands the number R[I-1] from 32 to 48 bits to prepare for the next step.

2. The 48-bit R[I-1] is XORed with K[I] and stored in a temporary buffer so that R[I-1] is not modified.

3. The result from the previous step is now split into 8 segments of 6 bits each. The left-most 6 bits are B[1], and the right-most 6 bits are B[8]. These blocks form the index into the S-boxes,

which are used in the next step. The Substitution boxes, known as S-boxes, are a set of 8 two-dimensional arrays, each with 4 rows and 16 columns. The numbers in the boxes are always 4 bits in length, so their values range from 0-15. The S-boxes are numbered S[1]-S[8].

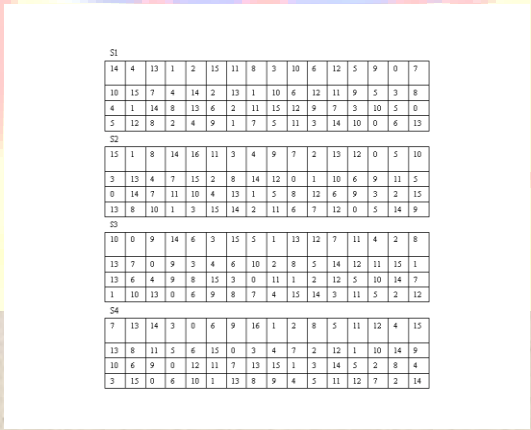
4. Starting with B[1], the first and last bits of the 6-bit block are taken and used as an index into the row number of S[1], which can range from 0 to 3, and the middle four bits are used as an index into the column number, which can range from 0 to 15. The number from this position in the S-box is retrieved and stored away. This is repeated with B[2] and S[2], B[3] and S[3], and the others up to B[8] and S[8]. At this point, we now have 8 4-bit numbers, which when strung together one after the other in the order of retrieval, give a 32-bit result.

5. The result from the previous stage is now passed into the P Permutation.

6. This number is now XORed with L[I-1], and moved into R[I]. R[I-1] is moved into L[I].

7. At this point we have a new L[I] and R[I]. Here, we increment I and repeat the core function until I = 17, which means that 16 rounds have been executed and keys K[1]-K[16] have all been used.

When L[16] and R[16] have been obtained, they are joined back together in the same fashion they were split apart (L[16] is the left-hand half, R[16] is the right-hand half), and the resultant 64-bit number is called the pre-output.



S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
10	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
5	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	16	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	14	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	16	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

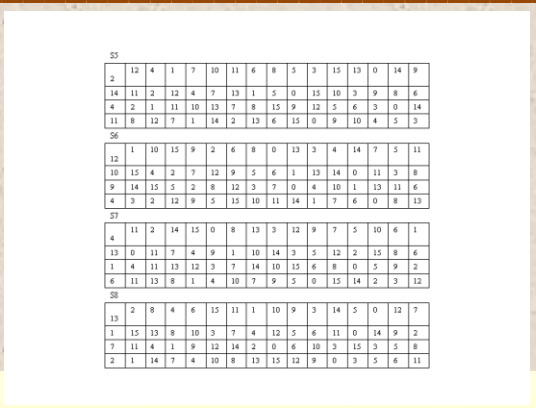


Table 3.1 Substitution boxes

DES Decryption:

As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the sub key is reversed.

Hash Function:

A hash value h is generated [3] by function h of the form

$$h=H(m)$$

Where m is the variable length message and H(m) is the fixed length hash value. We begin by examining the requirements for a hash function. Because hash functions are typically, quite complex, it is useful to examine that some very simple hash functions to get a feel for the issues involved.

A hash function H must have the following properties.

1. H can be applied to a block of data of any size.
2. H produces a fixed length output.
3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.
4. For any given value h, is computationally infeasible to find x such that H(x) =h. This is sometimes referred to in the literature as the one-way property.

5. For any given block x, it is computationally infeasible to find y ≠ x with H(y) = H(x). This is sometimes referred to as weak collision resistance.

6. It is computationally infeasible to find any pair(x,y) such that H(x) = H(y). This is sometimes referred to as strong collision resistance.

There are several similarities in the evolution of hash functions and that of Symmetric block ciphers. We have seen that the increasing power of brute-force attacks and advances in cryptanalysis have led to the decline in the popularity of DES and in the design of newer algorithms with longer key lengths and with features designed to resist specific cryptanalytic attacks. Similarly, advances in computing power and hash function cryptanalysis have led to the decline in the popularity of fusty MD4 and MD5, two very popular hash functions. In response, newer hash algorithms have been developed with longer hash code length and with features designed to resist specific cryptanalytic attacks. Another point of similarity is the reluctance to depart from a proven structure. DES is based on the Feistel Cipher, which in turn is based on the substitution-permutation network proposal of Shannon. Many important subsequent block ciphers follow the Feistel design because the design can be adapted to resist newly discovered cryptanalytic threats. If, instead, an entirely new design were used for a symmetric block cipher, there would be concern that the structure itself opened up new avenues of attack important modern hash function follow the basic structure. Again, this has proved to be a fundamentally sound structure, and newer designs simply refine the structure and add to the hash code length.

Our Approach to Secure Triple DES:

Demerits of existing process:

The strength of DES falls into loops holes. Eavesdroppers can easily hack the messages. The following are the reason for attacking.

1. **The use of 56 bit keys:** DES cracker machine was built in July 1998. A key search attack simply running through all possible keys.

2. **The nature of the DES algorithm:** There is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

3. **Timing attacks:** Timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryption on various cipher texts.

In order to keep the above strengths of DES concept, new secured approach is necessary. To protect the message from hacking, the following are to be considered by us.

New Approach:

Utilizing 168 bits key in Triple DES is safer than using 112 bits key. But key transformation becomes more complex while using 168 bit key. So we approach triple DES in a different way (i.e.) we use 168 bits key. First, keys are used by transmitting 112 bits. But all the 112 bits are not used directly in our process.

Consider the given 112 bit key is splitted into 56 bits each as K1 and K2. We then form K3 and K4 by encrypting K1 (by influencing K2 as key). Similarly K4 is formed by encrypting K2 (by influencing K1 as key). The keys K3 and K4 are then clubbed and a message digest of length 56 bits is produced for them. The message digest can be produced by using MD5 algorithm. But this MD5 algorithm produces only 128 bits message digest. So this 128 bit is stuffed to 56 bits. Now we obtain the third key for the encryption process.

Conventional EDE technique is used for encrypting the whole system First the encryption is performed by using K3. Next are decryption is performed by using the stuffed message digest. Finally the encryption is performed by using K4 Decryption also follows the conventions used as in DES decryption i.e. DES. Here encryption DES are performed by using K3, MD(K3,K4), K4 respectively. Similarly, decryption function is performed by using K3, MD(K3,K4), K4. For decryption DES, decryption is performed by encrypted K3. Next Encryption process is done through encrypted MD(K3,K4). Finally, decryption is performed by encrypted K4.

K1= 56 bits

K2 = 56 bits

K3 = Encrypted K1 using K2=56 bits

K4 = Encrypted K2 using K1=56 bits

MD=message digest of (K3 K4) =56 bits

Figure 4.1 represents the block diagram of new hybridizing approach for triple DES. The encryption DES is Encryption, Decryption and Encryption mode. The decryption DES is Decryption, Encryption and Decryption.

Encryption:

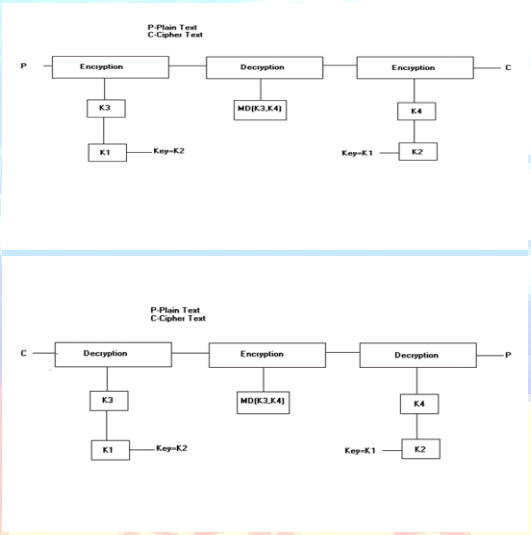


Figure 4.1: Secured Triple DES

Performance Analysis:

Since the keys are not used directly over here. This method gives no room for meet in middle attack. More over we are using message digesting technique which cannot be attacked .So on the whole, the system cannot be attacked. Further more if we calculate the risk factor the high percentage falls on the safer side of gauge. The core criterion that is to be observed is the splitting up of the keys into separate halves and their integration to form a message digest. This process helps one to entrance the security cordon a bit more.

The tiny edge which we get while forming message digest for keys is that, no system implementing message digestion technique has been breached until now. But still if we had created a message digest for the plain text that is obtained. Then there is a high profile chance of not getting the original plain text. This is the case because once a particular piece of information is wrapped up to form a message digest if cannot be retrieved back to its plain original form. So, in order to secure the data use form digests using the keys multiple key generations we spruce up the safety ration up a notch.

We must compare this type of approach with the other cryptosystems [4] like IDEA, Twofish and Blowfish. Even Blowfish is a symmetric block cipher that can be used as a drop in replacement for this Triple DES. In this method, one key is used so that we may split the key, make encryption by alternate keys. Message Digest Hash function has speed and secured message transmission among the networks. This type of processing is more or less like Advanced Encryption standard. Compare with all other cryptosystems, this type of approach is fast and secured one because encryption of keys by itself.

Conclusion:

Hybridization of message digest with triple DES is more secure and speed transmission than any other cryptosystem. If the security of this method proves to be adequate, it permits secure communications to be established without any difficulty. Encryption function is depending upon the data use from digest using the keys multiple key generations gives the safety. Computation will yield good confidence about the security. Using this concept, developing cryptosystem and implementation will do in future.

Reference:

- Andrew S. Tanenbaum, Computer Networks, PHI, 3rd Edition
- "Roaming PKI" *Information Security Magazine*, February 2000
- Stallings, W. *Cryptography and Network Security: Principles and Practice, Englewood Cliffs (NJ)* Patience Hall 1998
- Ferguson, N. and B. Schneier. *Practical Cryptography.* New York: John Wiley & Sons 2003

- Kessler, G.C. "Basics of Cryptograph and Applications for Windows NT." *Windows NT Magazine*, October 1999.
- *Kahn on Codes: Secrets of the New Cryptology.* New York: Macmillan, 1983.
- Bamford, J. (1983). *The Puzzle Palace: Inside the National Security Agency, America's most secret intelligence organization.* New York: Penguin Books.
- Bamford, J. (2001). *Body of Secrets : Anatomy of the Ultra-Secret National Security Agency from the Cold War Through the Dawn of a New Century.* New York: Doubleday.
- Barr, T.H. (2002). *Invitation to Cryptology.* Upper Saddle River, NJ: Prentice Hall.
- Bauer, F.L. (2002). *Decrypted Secrets: Methods and Maxims of Cryptology,* 2nd ed. New York: Springer Verlag.
- Denning, D.E. (1982). *Cryptography and Data Security.* Reading, MA: Addison-Wesley.
- Diffie, W., & Landau, S. (1998). *Privacy on the Line.* Boston: MIT Press.
- Electronic Frontier Foundation. (1998). *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. Sebastopol, CA: O'Reilly & Associates.
- Federal Information Processing Standards (FIPS) 140-2. (2001, May 25). *Security Requirements for Cryptographic Modules.* Gaithersburg, MD: National Intitute of Standards and Technology (NIST). Retrieved from http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
- Ferguson, N., & Schneier, B. (2003). *Practical Cryptography.* New York: John Wiley & Sons.
- Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering: Design Principles and Practical Applications.* New York: John Wiley & Sons.
- Flannery, S. with Flannery, D. (2001). *In Code: A Mathematical Journey.* New York: Workman Publishing Company.
- Ford, W., & Baum, M.S. (2001). *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption,* 2nd ed. Englewood Cliffs, NJ: Prentice Hall.
- Garfinkel, S. (1995). *PGP: Pretty Good Privacy.* Sebastopol, CA: O'Reilly & Associates.
- Grant, G.L. (1997). *Understanding Digital Signatures: Establishing Trust over the Internet and Other Networks.* New York: Computing McGraw-Hill.

- Grabbe, J.O. (1997, October 10). Cryptography and Number Theory for Digital Cash. Retrieved from http://www-swiss.ai.mit.edu/6.805/articles/money/cryptnum.htm
- Kahn, D. (1983). *Kahn on Codes: Secrets of the New Cryptology*. New York: Macmillan.
- Kahn, D. (1996). *The Codebreakers: The Story of Secret Writing*, revised ed. New York: Scribner.
- Kaufman, C., Perlman, R., & Speciner, M. (1995). *Network Security: Private Communication in a Public World*. Englewood Cliffs, NJ): Prentice Hall.
- Kessler, G.C. (1999, October). Basics of Cryptography and Applications for Windows NT. *Windows NT Magazine*.
- Kessler, G.C. (2000, February). Roaming PKI. *Information Security Magazine*.
- Kessler, G.C., & Pritsky, N.T. (2000, October). Internet Payment Systems: Status and Update on SSL/TLS, SET, and IOTP. *Information Security Magazine*.
- Koblitz, N. (1994). *A Course in Number Theory and Cryptography*, 2nd ed. New York: Springer-Verlag.
- Levy, S. (1999, April). The Open Secret. *WIRED Magazine*, *7*(4). Retrieved from http://www.wired.com/wired/archive/7.04/crypto.html
- Levy, S. (2001). *Crypto: When the Code Rebels Beat the Government — Saving Privacy in the Digital Age*. New York: Viking Press.
- Mao, W. (2004). *Modern Cryptography: Theory & Practice*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference.
- Marks, L. (1998). *Between Silk and Cyanide: A Codemaker's War, 1941-1945*. New York: The Free Press (Simon & Schuster).
- Schneier, B. (1996). *Applied Cryptography*, 2nd ed. New York: John Wiley & Sons.
- Schneier, B. (2000). *Secrets & Lies: Digital Security in a Networked World*. New York: John Wiley & Sons.
- Singh, S. (1999). *The Code Book: The Evolution of Secrecy from Mary Queen of Scots to Quantum Cryptography*. New York: Doubleday.
- Smith, L.D. (1943). *Cryptography: The Science of Secret Writing*. New York: Dover Publications.

- Spillman, R.J. (2005). *Classical and Contemporary Cryptology*. Upper Saddle River, NJ: Pearson Prentice-Hall.
- Stallings, W. (2006). *Cryptography and Network Security: Principles and Practice*, 4th ed. Englewood Cliffs, NJ: Prentice Hall.
- Trappe, W., & Washington, L.C. (2006). *Introduction to Cryptography with Codin Theory*, 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall.
- Young, A., & Yung, M. (2004). *Malicious Cryptography: Exposing Cryptovirology*. New York: John Wiley & Sons.