OPTIMIZED KEY MANAGEMENT SYSTEM IN CRYPTOGRAPHIC ALGORITHMS FOR DATA SECURITY

Thesis submitted in fulfillment of the requirement for the award of

the degree of

MASTER OF ENGINEERING (M.E.)

In

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

RAKESH KUMAR

Under the guidance of

Dr. AJAY KAKKAR

Assistant Professor

Electronics and Communication Engineering Department

Thapar University Patiala-147004 (India)

July-2014

DECLARATION

I hereby declare that the work, which is being presented in the dissertation, entitled "**Optimized Key Management System in Cryptographic Algorithm for Data Security**" in fulfillment of the requirements for the award of degree of Master of Engineering in Electronics and Communication Engineering submitted at Electronics and Communication Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the guidance of **Dr. Ajay Kakkar (Assistant Professor)**, Electronics and Communication Department and refers other research's work which are duly listed in reference section.

The matter presented in this dissertation has not been submitted in any other University/Institute for the award of degree.

Date: 15/7/2014

(Rakesh Kumar)
Roll No. 801261017

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 15/7/2014

(Dr. Ajay Kakkar)
Assistant Professor, ECED
Thapar University, Patiala

Countersigned by:

(Dr. Sanjay Sharma)
Professor and Head (ECED)
Thapar University, Patiala

(Dr. S.K. Mohapatra)
Dean, Academic Affairs
Thapar University, Patiala

ACKNOWLEDGEMENT

In my research work on Optimized Key Management System in Cryptographic Algorithm for Data Security for the present dissertation, I have received warm support, help and encouragement from my teachers for which I am deeply grateful.

I am beholden, first of all, to my supervisor, Dr. Ajay Kakkar (Assistant Professor, ECED) for patiently guiding me through the various stages of research. He has been very kind and illuminating indeed.

Thanks are due to Dr. Sanjay Sharma (professor and head, ECED) for his generous help with this thesis. I am also thankful to Dr. Kulbir Singh (Associate Professor) Electronics and Communication Engineering Department for his benign presence and helpful attitude.

I express my deepest gratitude to my parents for their blessing, unconditional love, support and encouragement. Their endless efforts have made a great contribution to all my successful endeavors in life.

(Rakesh Kumar)

TU, PATIALA

ABSTRACT

Data security is an important parameter in data communication. It is the prime requirement of all the organizations in order to keep their important information safe from hackers. There are various techniques, which are used to keep the data confidential from hacker. These are passwords, cryptography and biometrics. Passwords are not so good for this task due to their low entropy. Encryption is process of converting plaintext into cipher text. The strength of the cryptographic technique comes from the fact that no one can read the information without altering its content. Cryptography is the best solution for the above problem. In this report comparative study of various encryption algorithms has been done. From the Literature Survey, various observations and gaps have also been found. Objectives have been drawn from the observations and gaps. Optimization of keys for data encryption has also been done. Using dynamic keys simulation results have been achieved using MATLAB. Finally, comparison of our approach with existing algorithms has also been done on the basis of process time and working time. Future scope of the work has also been discussed in last chapter.

TABLE OF CONTENTS

Page No.

List of Figures

List of Abbreviations

PKC : Public key cryptography

DES : Data encryption standard

TDES : Triple Data Encryption Standard

AES : Advance Encryption Standard

CBC : Cipher Block Chaining

ECB : Electronic Code Book mode

OFB : Output Feed Back mode

KMS : Key Management System

IV : Initial Value

LSB : Least Significant Bit

MSB : Most Significant Bit

CHAPTER 1

INTRODUCTION

Cryptography has become one of the major methods for protection of data in all applications. It allows users to carry over the confidence found in the physical world to the electronic world. It enables the users to do business electronically without bothering of hackers. Earlier wax seals, signatures, and other physical mechanisms were used to assure integrity and data security. People communicate more electronically and upload their data in different fields. Increase of information transmitted through internet has increased the need of privacy and security of user's data [1].

Cryptography is the best method to avoid unauthorized access of data. It is the science of writing of data in secret code. It maps the original message in some random fashion which is unintelligible to unauthorized persons. Strength of cryptographic algorithm is defined from the number of attempts by hacker and time taken to break it [2].

It not only protects data from theft or alteration, but can also be used for user authentication. It is used to hide information and has its application in phone, fax and e-mail communication. It is the science of securing data; cryptanalysis is the science of analyzing and breaking secure communication. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis.

It is also used for information security issues such as electronic signatures, which are used to prove who sent a message. It processes data into unintelligible form, reversibly; so that, data can be recovered without data loss digitally.

Information security is concerned with the assurance of confidentiality, integrity and availability of information in all forms. It is the best solution for data security. It has two parts; a) Encryption and b) key. Encryption can protect communications and stored information from unauthorized access and disclosure.

1.1 Important Terms:

Plaintext: Data in its original form with the sender.

Ciphertext: Converted data in its unintelligible form [3].

Key: A block of fixed size of bits used to convert data in hidden form [3].

Encryption: The process of data converting in hidden form. It can include both encoding and enciphering [4].

Decryption: The process of recovering data in intelligible form [5].

Cipher: A pair of algorithm that perform encryption and decryption [6, 7].

Encipher: To convert plaintext into unintelligible form by means of a cipher system [8].

Encode: Convert plain text to equivalent cipher text by means of a code.

1.2 Security Goals

Cryptography involves two processes and is named as a) encryption and b) key management process. Every security system must provide some security functions that assure the secrecy of the system. These functions are called as goals. Using cryptography, many goals can be achieved and stated as:

a) Authentication [8]: It is the process of verifying the identity of the users before the communication in between them. It assures that communicating party is the one that it claimed to be.

b) Confidentiality: It means that only the authenticated people are able to interpret the message (data) content and no one else. It ensures that nobody can understand the received message except the one who has the decipher key. It ensures that system is secure [10].

c) Access control: Verifying if the user has the adequate permission to use the service. It prevents the unauthorized use of resources. It checks the conditions and restrictions for access to be occurred [11-12].

d) Integrity: It assures that data is not tampered or it is free from any modification in-between the end points [3].

e) Non-Repudiation [13]: This implies that neither the sender nor the receiver can falsely deny that they have sent a certain message.

f) Availability: Cryptographic model must be designed in such a way that, it must be available in case of any failure.

g) Accountability: All user actions that are security critical must be traceable back to the user. It stops the abuse of services as the malice activities are being traceable and can be punished.

1.3 Types of Cryptography

Cryptography performs encryption with the use of key. Based on the key management; there are two different categories of cryptography. a) **Private Key Cryptography**: It uses a single key for both encryption and decryption, b) **Public Key Cryptography**: It uses one key for encryption and another for decryption as shown in figure 1.1.

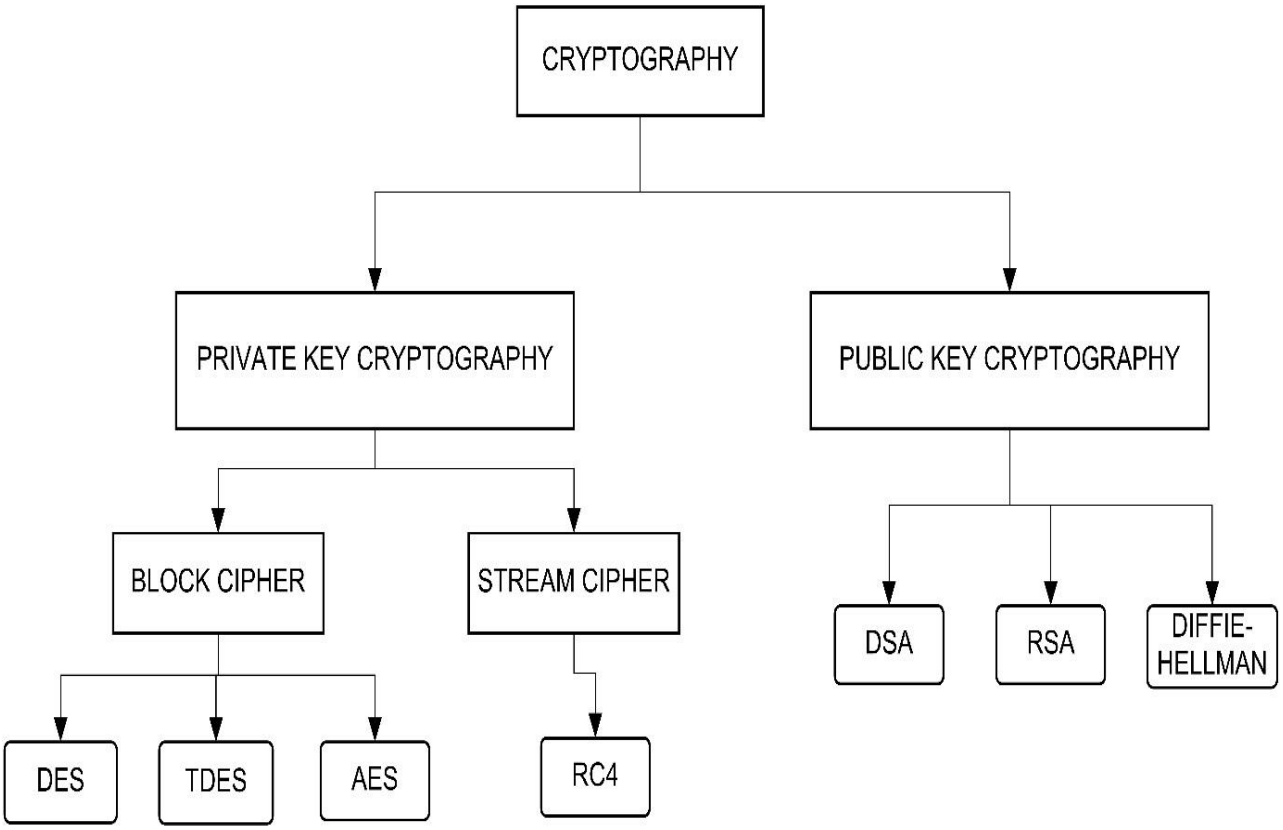


Figure 1.1: Classification of Encryption Systems

1. 4 Private Key Cryptography

In Private Key Cryptography (PKC) identical cryptographic keys are used for encoding of plain text as well as decoding of cipher text. The common key is a shared through secure channel between the participating parties [13, 14]. It is also called as secret key, single and shared key. If the symbols in the plaintext are alphabetic characters, replace one character with another. Symmetric Key Cryptography is generally used for long messages.

In this before enciphering both parties initially agree on a cryptosystem and then selects a key which will be used to for encryption and decryption as sown in figure 1.2. Using this approach data remains secure as long as the key is secure.

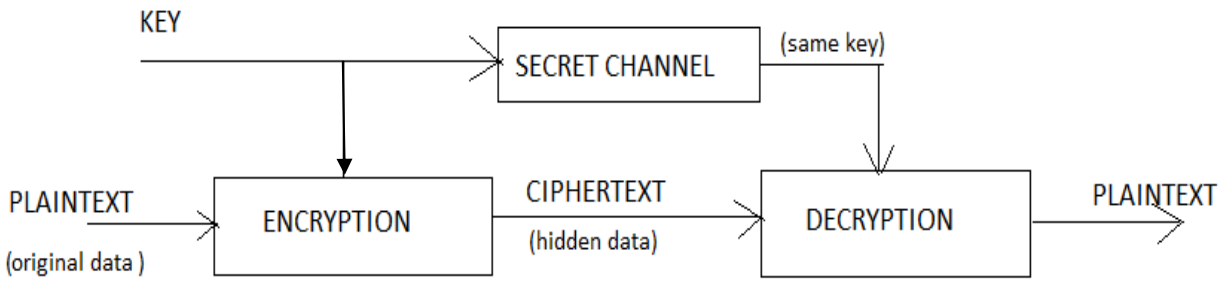


Figure 1.2: Secret key cryptography

Distribution of key is the major issue of this approach. Cryptographic algorithms differ with respect to the block size, the key size, the number of rounds the algorithm runs. Based on the input it takes to encrypt and decrypt data, Secret key cryptography schemes generally categorized as being stream cipher or block ciphers [14].

1.4.1 Block Cipher Algorithm

Block cipher algorithm [15] divides the plaintext in fixed length of blocks then by using a key it performs encryption on plaintext and produces the same length of block as a ciphertext as shown in figure 1.3. In this method all blocks are encrypted with the same key, which degrades security because each repetition in the plaintext will be a repetition in the ciphertext. To counter this issue, modes of operation are used to make encryption more reliable.

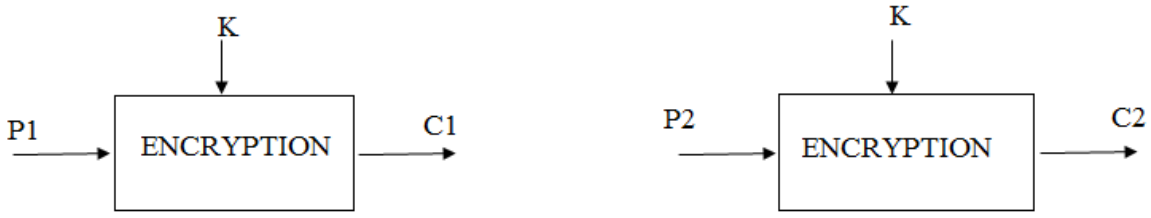


Figure 1.3: Block cipher in ECB mode

To provide proper scrambling one should not encrypt 2^(n/2) block with the same key using block cipher technique.

1.4.2 Stream Cipher Algorithm

Stream ciphers operate on a single bit at a time and implement some form of feedback mechanism so that key is constantly changing shown in figure 1.4. Stream algorithms are usually faster than block algorithms.

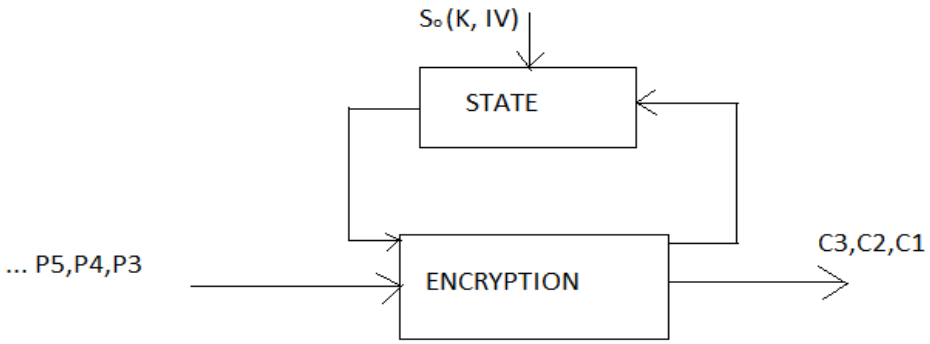


Figure 1.4: Stream cipher method

Where P3, P4, P5 are the plaintext and C3, C2, C1 are the ciphertexts. Synchronous stream ciphers generate the keystream in a fashion independent of the message stream but by using the same keystream generation function at sender and receiver. Block and stream cipher algorithm operate in different modes some of them are stated as:

• Electronic Codebook (ECB) mode: In this two identical plaintext blocks will always generate the same ciphertext block until the key is changed. It is susceptible to a variety of brute-force attacks [15]. It is suitable for parallel processing. It is simplest mode and error in one block transmission does not affect other block decryption shown in figure 1.5.

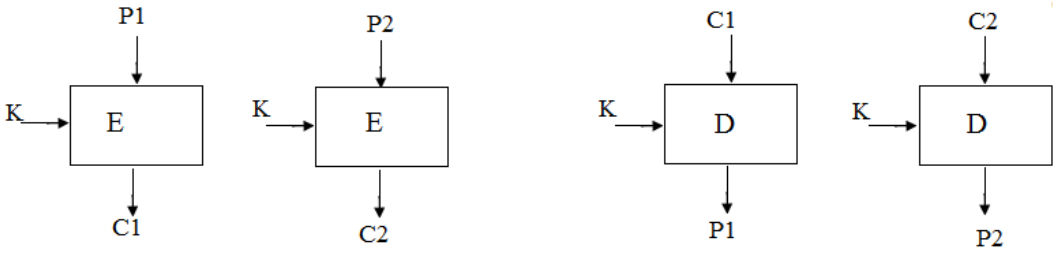


Figure 1.5: Block cipher encryption and decryption in ECB mode

• Cipher Block Chaining (CBC): It uses feedback mechanism to the encryption scheme. The plaintext is XORed with the previous ciphertext before applying to encryption. The first plaintext block is XOR-ed with a pre-defined Initial Value (IV) as

shown in figure 1.6. This scheme eliminates the drawback of repetition of blocks for same plaintext.

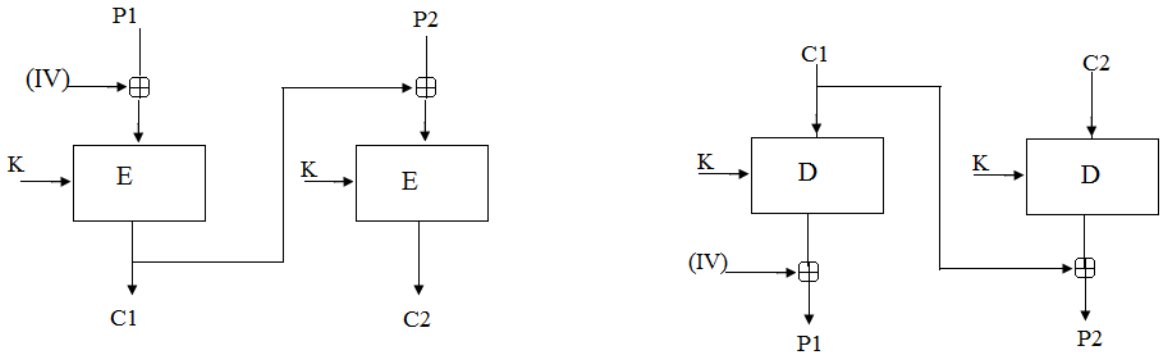


Figure 1.6: Block cipher in CBC mode

The disadvantage is that error in transmitting ciphertext make unrecoverable to next ciphertext also.

• Output Feedback (OFB) mode is a block cipher implementation conceptually similar to a synchronous stream cipher. Using IV a key stream is generated that is used to encrypt the block of plaintext. It prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that is independent of both the plaintext and ciphertext bit streams.

• Counter (CTR). In this mode a counter is encrypted by the block cipher and the result is XOR-ed with the plaintext block to obtain the ciphertext block. The counter is incremented before it is used in the next block. It allows the arbitrary position of data blocks to be processed.

Encryption: $Z_O = IV$, $Ci = Pi \oplus E_k(Zi)$, $Zi+1 = Zi + 1$.

Decryptiion: $Z_o = IV$, $Pi = Ci \oplus E_k(Zi)$, $Zi+1 = Zi + 1$.

• Cipher Feedback (CFB): It is a block cipher implementation as a self-synchronizing stream cipher. It allows data to be encrypted in units smaller than the block size. Previous result is XOR-ed with the new block of plaintext to obtain the new block of ciphertext.

A basic example of Symmetric Key Cryptography is a substitution cipher [17]. A substitution cipher substitutes one piece of information for another. It is frequently done by offsetting letters of the alphabet. For example, if we encode the word “KARNISAR”

using Caesar's key value of 3; therefore offset the alphabet so that the 3rd letter down (D) begins the alphabet. So starting with

ABCDEFGHIJKLMNOPQRSTUVWXYZ

And sliding everything up by 3, you get

DEFGHIJKLMNOPQRSTUVWXYZABC

Where D=A, E=B, F=C, and so on. Ciphertext = "NXUQLVXU"

Similarly one can have their own dynamic value, for example instead of number 3, if we take number 8; then offset the alphabet so that 8th letter down (I) begins the alphabet and ciphertext changes as:

Ciphertext = "SIZVQAIZ".

1.4.3 Symmetric vs. Session Key

In session key; the symmetric key can be changed every time in communication between two parties. It is randomly generated and valid for only one session. If an attacker gets the session key, he can decrypt only the messages for a particular session. If both parties always used the same key for all sessions, the attacker would be able to decrypt all messages encrypted with this key.

1.4.4 Scalability and Secure Key Distribution

Scalability is the main problem with symmetric ciphers. If there are x people who want to communicate with each other, they need (x-1) different keys to establish separate and confidential communication channels. Another problem is secure key distribution. The security of the system is broken if a man-in-the-middle can get the key while it is being transmitted from one user to another.

1.5 Key Management in Private Key Encryption

The key management [20] [23] is challenging task, as a unique secret key is used for peer to peer connection. It is very fast and useful for encrypting data. Key management should be able to generate a secret key between two parties. The aim is to store it, to prove the authenticity of the keys and of the communicating parties. As soon as a key has been generated it must be prevented safe. Key management system is building a database which is used for storing the key. A reliable session key has to be implemented for an effective encryption at the data transfer.

1.6 Advantages and disadvantages of Symmetric Key Cryptography

In symmetric key cryptography sender and receiver only have to specify the shared key in the beginning and then they can begin to encrypt and decrypt messages between them using that key. It has several advantages and disadvantages in compare to other techniques as:

Advantages:

a) Simple: This encryption scheme is easy to carry out. Initially all users have to do specify and share the secret key and then begin to encrypt and decrypt messages.

b) Encrypt and decrypt your own files: If one uses encryption for messages or files which he alone intend to access, there is no need to create different keys. Single-key encryption is best for this.

c) Fast and suitable for longer data: Symmetric key encryption is much faster than asymmetric key encryption. Due to less computation complexity it is suitable for longer messages.

d) Uses less computer resources: Single-key encryption does not require a lot of computer resources when compared to public key encryption.

e) Prevents widespread message security compromise: A different secret key is used for communication with every different party. If a key is compromised, only the messages between a particular pair of sender and receiver are affected. Communications with other parties are still secure.

Disadvantages:

a) Need for secure channel for secret key exchange: Sharing the secret key in the beginning is a problem in symmetric key encryption. It has to be exchanged in a way that ensures it remains secret.

b) Member of keys: A new shared key has to be generated for communication with every different party. This creates a problem with managing and ensuring the security of all these keys.

c) Origin and authenticity of message cannot be guaranteed: Since both sender and receiver use the same key, messages cannot be verified to have come from a particular user. This may be a problem if there is a dispute between parties.

1.7 PUBLIC KEY CRYPTOGRAPHY

PKC involves two keys which are mathematically and does not allow someone to easily determine the other key. It uses a pair of different key, known as asymmetric cryptography [28]. In PKC, one of the keys is designated the public key and may be advertised as widely as the owner wants. The other key is designated as private key and is never revealed to another party.

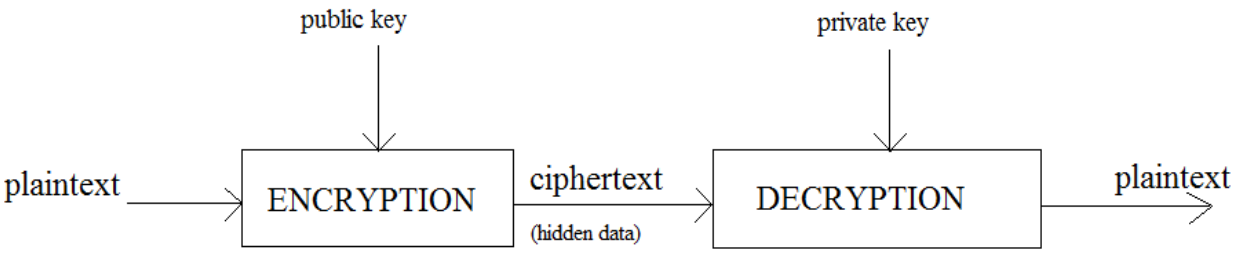


Figure 1.7: Public key cryptography

Public key used to encrypt data, and corresponding private, or secret key used for decryption. It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information [30].

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Some examples of public-key cryptosystems are RSA (Ron Rivest, Adi Shamir, and Leonard Adleman), Diffie-Hellman (inventors), and DSA, the Digital Signature Algorithm. As conventional cryptography was the only available means for sharing secret information, the expense of secure channels and key distribution relegated its use only to those who could afford it, such as governments and large banks. This scheme made it for others also. The problem of SKC's key distribution was also solved by public key cryptography.

1.7.1 Key Management in PKC

It allows a person to send a message that can only be read by the intended receiver, without having a need for the sender and receiver to agree on a secret key. Key used for

encryption is different from the key used for decryption. In PKC, the public and private keys are mathematically related, but still it is very difficult to derive the private key given only the public key; however, deriving the private key is always possible given enough time and computing power. This makes it very important to pick keys of the right size; large enough to be secure, but small enough to be applied fairly quickly. In addition to the convenience of key management [45, 47, 48] for encryption; PKC also provides a means to implement digital signatures. The separation of public and private keys is required to allow users to sign their data.

The secret key provides the link between the public key and the individual, and will remain a valid link only if the user properly maintains the secrecy of the private key. If for some reason a user's secret key for a digital signature scheme is compromised, the public key may need to be revoked. If it is known when the private key was compromised, then there is no need to invalidate all of the documents that were signed prior.

1.7.2 Advantages and disadvantages of Public Key Cryptography

The public key is made publicly available and is used to encrypt messages by anyone who wishes to send a message to the person that the key belongs to. The private key is kept secret and is used to decrypt received messages. RSA is an example of asymmetric key encryption.

Advantages:

a) Convenience: It solves the problem of distributing the key for encryption. Everyone publishes their public keys and private keys are kept secret [49].

b) Provides for message authentication: PKC allows the use of digital signatures which enables the recipient of a message to verify that the message is truly from a particular sender.

c) Detection of tampering: The use of digital signatures in public key encryption allows the receiver to detect if the message was altered in transit. A digitally signed message cannot be modified without invalidating the signature [49].

d) Provide for non-repudiation: Digitally signing a message is akin to physically signing a document. It is an acknowledgement of the message and thus, the sender cannot deny it.

Disadvantages:

a) Public keys must be authenticated: No one can be absolutely sure that a public key belongs to the person it specifies and so everyone must verify that their public keys belong to them.

b) Slow: Public key encryption is slow compared to symmetric encryption. Not feasible for use in decrypting bulk messages.

c) Uses up more computer resources: It requires more computer supplies compared to single-key encryption.

d) E**ntire communication compromise is possible:** If an attacker determines a person's private key, his or her entire messages can be read.

e) Loss of private key may be irreparable: The loss of a private key means that all received messages cannot be decrypted.

1.8 Digital Signatures

A major benefit of public key cryptography is that it provides a method for employing digital signatures. It enables the recipient of information to verify the authenticity of the information's origin, and also verify that the information is intact. Thus, public key digital signatures provide authentication and data integrity [47, 49]. A digital signature also provides non-repudiation, which means that it prevents the sender from claiming that he or she did not actually send the information. A digital signature serves the same purpose as a handwritten signature. However, a handwritten signature is easy to counterfeit. A digital signature is superior to a handwritten signature as it is nearly impossible to counterfeit. It also attests to the contents of the information as well as the identity of the signer.

1.9 Hash functions

Hash functions, also called message digests and one-way encryption, are algorithms that does not use any key. A fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Hash algorithms [50] are used to provide a digital fingerprint of a file's content often used to ensure that the file has not been altered by an intruder.

Public key cryptosystem is slow, and it produces a large volume of output data i.e. at least double the size of the original information. An improvement on this scheme is the

addition of a one-way hash function in the process. A one-way hash function takes variable-length input as a message of any length and produces a fixed-length output. Hash function also ensures that, if the information is changed by just one bit produces an entirely different output value is produced.

1.10 Significance of key length and strength of SKC vs PKC

The effectiveness of protection depends on a number of issues such as cryptographic key size, protocol design, and password selection. Generally, the strength of encryption algorithm depends on difficulty in getting the key, which depends on both the cipher used and the length of the key. Key length is chosen as the first parameter for specifying cryptographic algorithm. Key length is measured in number of bits. If a key is too small, or if a protocol is badly designed, then the protection fails and improper access can be gained.

An attacker tries to map the relation between plaintexts and ciphertexts by observing it for a number of encryption [51]. There should be a large variation in ciphertext with a small change in key bits, known as avalanche variation. Every extra bit added doubles the number of possible key. For longer key, with brute force attack more number of attempts will be required to break it.

Strength: Generally asymmetric encryption schemes are more secure because they require both a public and a private key. But it's not a compulsion. Symmetric cryptography and asymmetric cryptography are two different kinds of cryptographic tool. Each one can be either weak or strong. The strength of encryption is determined by the key size. Asymmetric algorithms require large keys than symmetric to provide same degree of security.

AES is more secure against cryptanalytic attacks than 512-bit RSA, even though RSA is asymmetric and AES is symmetric. 4096-bit RSA is more secure against cryptanalytic attacks than 40-bit RC4, even though RC4 is symmetric and RSA is asymmetric. Conventional 80 bit key has the equivalent strength of a 1024 bit public key. A conventional 128- bit key is equivalent to a 3000 bit public key.

Generally, the more random the key, the larger the key, the secret the algorithm and the safer the key distribution system provides a better encryption.

1.11 Hybrid Cryptography

Symmetric Key Cryptography is easy to compute and the Public Key Cryptography is more secure compared to symmetric but it is slow. By combining the advantages of these two cryptographic methods, the level of security can be enhanced which is called hybrid cryptography [51], used now a days. In Public Key Cryptography, the end to end encryption does not provide a provision for aggregation on intermediate nodes because it is unaware about the private key of sink node. To achieve adequate security both symmetric and asymmetric cryptography are combined. An asymmetric encryption is used to exchange the secret key, symmetric encryption is then used to transfer data between sender and receiver i.e. asymmetric approach is used to confirm the identity of a communication partner and symmetric key is then used to perform encryption on the actual data [51, 64]. In hybrid cryptography the key is created with the help of public key cryptography, the generated key is fed to AES (Symmetric Key Cryptography) for end to end encryption to achieve the confidentiality.

1.12 Thesis Organization

This thesis includes five chapters. An outline of each chapter is given below:

The 1st chapter gives an introduction to the cryptography and its types. Key management techniques, cryptographic definitions and some of cryptographic used for security purpose in communication. Significance of key length has also been discussed in this chapter.

Chapter 2 is dedicated to the literature survey. The research papers which are relevant to this work are discussed here. Observations and objectives have been derived in this chapter.

Chapter 3 presents a study of the symmetric key cryptography and its structure. In this chapter encryption, decryption and simulation programmed have also been discussed in detail.

Chapter 4 includes the simulation, carried out using MATLAB. The discussion of our new algorithm and its results presented. Finally, comparison of our approach with existing algorithm has been done.

Chapter 5 shows the conclusion of this thesis with results and suggestions for hopeful future work and modification in this algorithm.

CHAPTER 2

LITERATURE SURVEY

This chapter involves the work done by the various researchers in the field of cryptography. From the literature survey, few observations have been drawn and are stated at the end of this chapter. Finally from these observations, objectives of this work have also been derived.

Sarker and Parvez [1] proposed a cost effective symmetric key cryptographic algorithm for small amount of data. For a very minimal amount of data DES, AES and IDES were cost effective therefore these were not designed for small amount of data. They proposed an algorithm which was designed in a quite simple manner and involves all the security issues. It was used for both encryption and decryption but as public key cryptography was more secured; therefore secret key cryptography not fulfills the security issue completely.

Keromytis *et al.* [2] discussed the design procedure of the open Berkeley Software Distribution (BSD) cryptographic framework. Cryptographic transformations had a fundamental building block in many security applications and protocols. To improve performance of Open BSD Cryptographic Framework (OCF), a service virtualization layer implemented inside the kernel, which provided uniform access to accelerator functionality by hiding card-specific details behind a carefully-designed API. They also evaluated the impact of the OCF in a variety of benchmarks. By measuring overall system performance and aggregate throughput; they concluded that the OCF was extremely efficient in utilizing cryptographic accelerator functionality. It attains 95% of the theoretical peak device performance.

Ahmed [3] proposed a new symmetric key generation algorithm using sum of subset problem. They focused on information security which was the process of protecting information and it protects its availability, privacy and integrity. It increased the strength of the key while keeping the size of the key optimized. Hence encrypted data was more difficult to crack by a brute force technique and overhead of data encryption was also comparable to existing algorithms. It could be used for symmetric encryption of data while maintaining the integrity and security of the data.

Rudinger and Finger [4] worked on the designing and side channel vulnerability of Differential Power Attack (DPA). It was proved that the careful selection/design of

algorithms can increase the security of implementations of algorithms. The ideal properties of an algorithm from the theoretical viewpoint of complexity were a) Large block length and key bit dependency for any intermediate results of the cipher, b) Complex key scheduling; c) Large key length. These algorithmic countermeasures work independently and to conventional side channel countermeasures and could be important because the increase in theoretical complexity is independent of the channel on which the attack is performed. It makes it independent from conventional DPA attack countermeasures.

Yan and Xiao [5] studied block algorithms which were implemented on hardware in information security system. Due to its block cipher decrypted speed, high security strength, low cost and easy to realize, it was widely used for data security. They also introduced the hardware realization of Rijndael encryption algorithm; due to the limited time and ability this work is still very limited in the information security and confidentiality measures and still need to do the realization and the performance of the algorithm of optimization and more in-depth research.

Anand *et al*. [6] explored identity-based cryptography techniques and applications. They reviewed the identity based encryption applications in the field of various networks as ad-hoc networks. The scheme also used in mobile networks and other wireless networks. They also discussed that under what parameters identity based cryptography was used with its benefits and limitations. The main limitation was that the available methods were restricted to fixed output block, which was a trace for crackers.

Fakhar and Shibli [7] worked on management of symmetric cryptographic keys in cloud based environment. Cloud computing provides innumerable benefits to its customers but it fails to solve information security concerns especially in public cloud. Sensitive data storage on cloud platform was challenging while adopting cloud services for data storage. Cryptographic keys were based upon sensitive data and required cloud platform in different cases.

Murphy *et al.* [8] worked on hardware-software implementation of public key cryptography used for Wireless Sensor Network (WSN). Protocols were used to ensure synchronization of keys between the devices in a network. These protocols required a significant communication and suffered from overhead. Using a hardware/software code sign approach, they had successfully mapped a public key cryptosystem based on Rabin's

scheme. Their implementation was focused on efficient architectures which executes the public key algorithms using minimal resources. The limitation of such a cryptosystem was that they not provide the guarantee of confidentiality for the session keys.

Krawczyk [9] proposed the order of encryption and authentication for protecting communications. It composes symmetric encryption and authentication to build secure channels for the protection of communications over insecure networks. They also proved that any secure channels protocol designed to work with any combination of secure encryption (against chosen plaintext attacks) and secure MAC must use the encrypt-then-authenticate method. It was prone to ciphertext forgery attack when encryption of the plaintext and MAC tag are done separately.

Tannous *et al.* [10] presented new side channels that leak password information during windows keyboard processing of password. Side channels were typically viewed as attacks which leak the cryptographic keys during cryptographic algorithm processing. They also proved that (a) side channels were not eliminated by removing accurate clocks or hardware cache mechanisms (b) side channels were of continued concern for computer security as well as cryptographic processing.

Karandikar *et al.* [11] proposed an effective key management approach for Differential Access Control (DIF-AC) in dynamic environment. It was based on Secure Group Communication (SGC) key management and Secrecy Despite Compromise (SDC). It scales better in a dynamic scenario when users change subscriptions or service providers add or revoke resources. Comparing with other schemes it was compact, efficient, practical, and generic. They also proposed a novel approach of keys management to enforce DIF-AC in highly dynamic environments, based on secure group communication framework.

He *et al*. [12] suggested a self-contained public key management scheme for critical wireless Adhoc networks named as SMOCK. It required significantly less key storage space than other and was capable to resist the Sybil attack. It achieved zero communication overhead for authentication and fulfills the secure communication requirement in terms of integrity, authentication, confidentiality, non-repudiation, and service availability. Small numbers of cryptographic keys were stored off-line at individual nodes before they were deployed in the network. To provide good scalability in terms of number of nodes and storage space, they utilized a combinatorial design of

public-private key pairs. It means nodes combine more than one key pair to encrypt and decrypt messages.

Devi [13] discussed the importance of cryptography in network security. Network security and cryptography was used to protect information in digital form and to provide security services and still has their importance in real time application. The purpose of a digital signature was to provide a means for an entity to bind its identity to a piece of information. Author discussed some common attacks on digital signature and stated that RSA signature is most practical versatile scheme due to its data integrity and non-repudiation.

Abiachi *et al*. [14] provides a competitive study of cryptography techniques over block cipher. Cryptography process seeks to distribute an estimation of basic cryptographic primitives across a number of confluences. It reduces the security assumptions on individual nodes, which established a level of fault-tolerance, opposing to the node alteration. It obtained a high security during the encryption and decryption process. It was based upon text contents and simplified the key management process. The complexity of this block cipher cryptographic model does not allow a graph to exchange the data in secured means.

Ren and Harnsciences [15] proposed generalized ring signatures. In a ring signature set of possible signers are specified. So the verifier was unable to tell which member actually produced the signature. They also introduced a generalized multi signer ring signature scheme to increase the level of confidence or enforce cross-organizational joint message singing in which all ring members could share the same prime number and all operations could be performed in the same domain. It was convertible ring signature that enabled the actual message signer to prove to a verifier that only that user is capable of generating the ring signature. It could achieve unconditional signer ambiguity and was secure against adaptive chosen-message attack in the random oracle model.

Lan *et al.* [16] proposed a Random Number Generator (RNG) for low power cryptographic applications. It was widely used in cryptographic systems as the cryptographic keys generator. These keys were most important component in the system because the security of the cryptographic system relies entirely on its quality. They also presented the good statistical quality and low energy consumption RNG including a serial-to-parallel shift register, a 32-bit register and a pseudo random number generator

(PRNG) module which could be suitable for low-power, flexible cryptographic applications. They also suggested that it could be implemented completely in digital circuit and required no external components.

Delgrande *et al*. [17] introduced a declarative, logical approach for the representation and analysis of cryptographic protocols which were usually specified in an informal, ad hoc language, with crucial elements, such as the protocol goal, left implicit. They observed that many proofs of protocol correctness rely on the assumption that honest agents do not perform actions that compromise secret information; but it was not always clear which actions were likely to do so. They specified axioms that restrict honest agents from performing them. Their prototype verification software took a protocol specification, translated it into a high-level situation calculus (Golog) program, and outputs any attacks that could be found.

Lo *et al.* [18] proposed an effcient key assignment scheme for access control in a large leaf class hierarchy. It ensured that the security of the predecessor could not be revealed by any unauthorized successors from the violation of access policy. It offered a perfect reduction on the number of primes, which resulted in a notable efficiency improvement. It uses the lowest amount of key regeneration when the leaf class was added or removed. It provides the highest efficiency compared to other schemes and maintains the essential and sufficient security in a large POSET (partially ordered set) hierarchy.

Tiri [19] worked on side-channel attacks in cryptographic algorithms which were usually strong against mathematical attacks. They showed that without expensive equipment or intrusive monitoring, these attacks bypass the mathematical complexity. They find the cryptographic key by observing the power consumption and/or the execution time variations of the device in normal operation mode. It facilitates the observation of the information leakage and discussed mitigation strategies and identified opportunities for future research.

Khaing and Aung [20] discussed secured key distribution scheme for cryptographic key management system that limits amount of cipher text available to attackers and also limit the exposure in event of key compromise. It was capable of self-adaptive key establishment for large-scale users as well as reduces the computational complexity. They also discussed time interval rekey process which was more appropriate than individual rekey process, if the system had a dynamic access patterns. The limit of this scheme was

that performance improvement could be achieved at the expense of delayed key renewals along the departed user path in the key tree scheme.

Bhatele *et al.* [21] suggested a novel approach for the designing of new hybrid security protocol architecture. They introduced a new security protocol for on-line transaction which could be designed using combination of both symmetric and asymmetric cryptographic technique known as hybrid cryptography. This protocol serves three very important cryptographic primitives - integrity, confidentiality and authentication. The symmetric cryptographic algorithms are fast as compared to asymmetric cryptographic algorithms; therefore when both are used together in a proper way, the result provides high security with fast speed. They tried to encapsulate all the developments introduced in the designing of new security protocol for on-line transaction. It was more immune against the square attacks because of the inclusion of AES algorithm and also provides short response time.

Sakiyama *et al.* [22] worked on information theoretic approach used for optimal Differential Fault Analysis (DFA). They showed a comprehensive analysis of differential fault analysis attacks AES from an information-theoretic perspective. Injecting faults into cryptosystems was categorized as an active attack where hackers induce an error in operations. This was done to retrieve the secret internal information concerned with the secret keys of ciphers. They considered DFA attacks as equivalent to a special kind of passive attack where attackers could be obtained from leaked information. The main limitation of their work was that the analysis for it was carried not by signing the noise component.

Divya *et al.* [23] presented a Key management scheme based on Elliptic Curve Cryptography (ECC) which works well against non-differential side channel attack. It reduced storage space requirement, communication overhead and provided security using unified addition formulae for point multiplication in ECC. It also ensured saving of energy consumption for point multiplication. The probability of compromising other node with captured information was zero when one node was compromised because the keys were independent to each node. The limitation was that it could not be applied to all types of elliptic curve.

Rangari *et al.* [24] worked upon an enhanced symmetric key cryptography algorithm to improve data security which was essential for block cipher method. It takes less time for

providing security for bully files. It becomes very tough to break the encryption algorithm without knowing the exact key due to internal key generation with the reference of entered key. The proposed method for both encryption and decryption could be applied for any type of public application for sending confidential data. It could be also be useful to send internal key to the sender using another secured path to the receiver. It prevents data from attackers and claim for less time complexity for large data files.

Harn and Lin [25] worked on a cryptographic key generation scheme for multilevel data security. They observed that there were two main problems associated with this scheme. a) A large value associated with each security class needs to be made public. b) New security classes were not permitted and could be added into the system once all the security keys were issued.

Eschenauer and Gligor [26] worked on a key-management scheme for Distributed Sensor Networks (DSN). They include sensor nodes with limited computation and communication capabilities. They were dynamic allows addition and deletion of sensor nodes after deployment to grow the network or replace failing and unreliable nodes. They also suggested that DSNs were deployed in hostile areas where communication was monitored and nodes were subjected to capture. The key-management scheme for security and network connectivity characteristics was also discussed. It was scalable and flexible; trade-offs could be made between sensor-memory cost and connectivity. Design parameters could also be adapted to fit the operational requirements of a particular environment.

Villalba *et al*. [27] introduced a secured extension to the Optimized Link State Routing (OLSR) protocol. This study presents an extension of OLSR named as COD-OLSR, which provides security for OLSR in the case of incorrect message generation. It takes into account the current topology of the sending node, so that the receiver of the message could verify the integrity of control messages. It reduces the overhead work and confirming security of model. The behavior of COD-OLSR against different attackers in a variety of situations was also evaluated. The simulation results proved that COD-OLSR adds a slight overhead to OLSR and barely affects performance.

Uhsadel *et al.* [28] proposed hardware-software co-design with a public key cryptography application. Their aim was to prove that there were many alternative solutions in the design space and to teach the fundamental concepts of hardware-software co-design. It

was attractive for pedagogic purposes because its complex arithmetic and large word lengths made it difficult to realize in software on an embedded microcontroller. But the alternative of a pure Application-Specific Integrated Circuit (ASIC) application was also not a satisfactory solution, as this lacks the flexibility to support multiple public key applications.

Biri *et al.* [29] proved that there was a trade-off between identity-based and certificateless cryptography for online application. Identity Based cryptography (IBC) has a special property that user's public key related directly to their identity. IBC suffers from key escrow due to the fact that the user private key was generated by an external entity called the Private Key Generator (PKG). To resolve this problem, the certificateless cryptography CL-PKC was proposed. This mechanism uses a different method to generate the public key. Although CL-PKC resolves the IBC problem, it removes the IBC attractive advantage of easy public key generation. In order to tackle these problems they proposed a new solution which merges these two cryptographic systems into only one system. It provides a self-generated-certificate along with the important advantages of IBC such as keeping user identity related to a public key.

Salam *et al*. [30] proposed a key pre-distribution scheme for WSN using public key cryptography. Efficient bootstrapping of secure key was the critical factors to ensure security in WSN. It provided node to node authentication which provides resistance against node replication and with nominal increase in overhead. It also provides revocation of compromised node.

Wang *et al.* [31] worked on single and multi-core configurable AES architectures for flexible security. Each AES processor provides block cipher which had a key expansion design. In this multi core architecture; the memory controller of each AES processor was designed for the maximum overlapping between data transfer and encryption. It reduces interrupt handling load of the host processor. This design could be applied to high-speed systems; it had independent data paths and also reduces the I/O and bandwidth problem.

Shemaili *et al.* [32] worked on a new lightweight hybrid cryptographic algorithm for the World Wide Web (WWW). The internet world was continuous revolutions from the World Wide Web and the mobile internet to the Internet of Things (loT). Radio Frequency IDentification (RFID) and Wireless Sensors (WS) were technologies that could be used to create the loT world. They addressed some lightweight ciphers and came

up with a new algorithm which was fit to low computation devices such as RFID and WS. It holds a very sensitive data which was related to the physical world such as the names or places of people.

Neal Koblitz [33] proposed an elliptic curve cryptosystems based on elliptic curves over finite fields of public key cryptosystems. It uses the multiplicative group of a finite field and was treated as more secured.

Puhan and Ho [34] proposed a new binary document image watermarking for secured authentication using perceptual modeling. They also proposed a new perception based watermarking algorithm in binary document images for authentication. The black and white pixels in such simple images were noticeable visual distortion. The reversible property of the curvature-weighted distance difference (CWDD) measure was used towards designing a new authentication watermarking algorithm. Hence, the possibility of any undetected modification to the watermarked image was removed.

Moebius *et al.* [35] presented a model-driven approach to generate a formal specification from a Unified Modeling Language (UML) model. It was used to generate specifications for verification and executable code. This model-driven approach was focused on security-critical applications which were based on cryptographic protocols. The formal specification was based on abstract state machines and algebraic specifications. It allowed to formulate and to prove application-specific security properties.

Lalithamani and Soman [36] worked upon the generation of irrevocable key for cryptography based on cancelable fingerprints. Incorporation of biometrics with cryptography is a proficient solution for random articles. The fingerprints were utilized to extract the minutiae points that were altered, in an efficient manner to acquire transformed points. The transformed points were used to produce the cancelable templates which were in turn utilized for the extraction of irrevocable keys. It was extremely unfeasible to obtain cancelable fingerprint templates and original fingerprints from the generated key since the cryptographic key produced was irrevocable.

Khanna *et al.* [37] proposed new symmetric key cryptographic algorithm using combined bit manipulation and Message Security Assist (MSA) encryption algorithm. It allowed the multiple encryptions and decryptions. It was applied in serial manner to increase the strength of the encryption and decryption. It was assumed that it was almost impossible to break the encryption algorithm without knowing the exact key matrix. Another advantage

of the method was that one could apply this to any other standard algorithm such as DES, AES or RSA. It was a block cipher method and could be applied to encrypt data in sensor network or in mobile network.

Traynor *et al.* [38] proposed an efficient hybrid security mechanism for heterogeneous sensor networks. They observed that a probabilistic unbalanced distribution of keys throughout the network while leverages the existence of a small percentage of more capable sensor nodes. It also reduced the consequences of node compromise.

Gope *et al.* [39] proposed a new block cipher cryptographic symmetric key algorithm named TACIT encryption technique for secure routing. It used an independent approach with suitable mathematical which was assumed to be computationally secured. Key distribution system was being applied on a secure policy based routing. It was limited to conversion of text file.

Rivest *et al*. [40] proposed a method for obtaining digital signatures and public-key cryptosystems whose security rests in part on the difficulty of factoring large numbers. An encryption method was presented which publicly revealing an encryption key. It has two important consequences; a) Couriers or other secure means were not needed to transmit keys, since a message could be enciphered using an encryption key publicly revealed by the intended recipient, only particular user could decipher the message. b) A message could be signed if using a privately held decryption key. Anyone could verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forgotten and signers cannot later deny the validity of their signature. The technique was used in e-mail and secured electronic funds transfer systems.

Davis [41] worked on the Data Encryption Standard (DES) in perspective with other computer security measures. It was applied to federal computer systems either before or coincident. They also observed that the environment surrounding and the history of the DES was be used to detect the crucial information about the keys; therefore there was a need to use additional standards to be developed within the computer security program.

Dutta *et al.* [42] proposed constant storage self-healing key distribution with revocation in WSN which was less complex. This scheme enabled a large group of users to establish a session key dynamically over an unreliable and lossy wireless network. It uses a different and more efficient self-healing mechanism compared to other schemes. It was

unconditionally secure and could achieve both forward and backward secrecy. The proposed self-healing key distribution was not restricted to total sessions in initial phase.

Owor and Hamilton [43] proposed an elliptical cryptographic algorithm for RF wireless devices. It is an asymmetric cryptographic algorithm based on the elliptical curve cryptographic approach. It makes use of an orthogonal frequency division multiplexing based upon RF wireless system which uses planner matrix for encryption. The main advantages of proposed algorithm were fast and efficient implementation of prime number factoring, logarithmic transformation and increased difficulty of finding inverse solutions.

Pletka and Cachin [44] discussed cryptographic security for a high-performance distributed file system. Cryptographic file systems mitigated the danger of exposing data by using encryption and integrity protection methods and guaranteed end-to-end security for their clients. They described a generic design for cryptographic file systems and its realization in a distributed Storage-Area Network (SAN) file system. Key management was integrated with the meta-data service of the SAN file system. It supported file encryption as well as integrity protection through hash trees. But overhead was noticeable for some artificially constructed use cases.

Azarderakhsh *et al.* [45] et al. presented a key management scheme for clustered WSNs, which used both public and symmetric key cryptography. In this each node requested a session key from the gateway to establish a secure link with its neighbors instead of preloading a large number of keys into the sensor nodes. It used public key cryptography for giving a session key to a sensor node, which had already requested for a session key through its potential neighbors. It suffered from key revocation problem.

Vignesh *et al.* [46] worked on the advancements of quantum and the versatility of classical cryptography. They also presented some of the quantum's theoretical weaknesses such as lack of digital signatures along with other real time implementation problems. They pointed out the vulnerabilities of transmission through quantum channel. This restriction was basically due to the fact that algorithms could not be implemented in QC without sacrificing on security.

Li *et al.* [48] presented a scalable key management and clustering scheme for secure group communications in adhoc and sensor networks. A scalable key management and clustering scheme for secure group communication in adhoc and sensor network was also

used. The scalability problem was solved by partitioning the communicating devices into subgroups. The Distributed Efficient Clustering Approach (DECA) provided robust clustering to form subgroups. It was energy efficient and resilient against node mobility. Comparing with most other schemes, it was extremely scalable and efficient, provided more security guarantees, and is selective, adaptive and robust. It terminates fast and has low time complexity and generates non-overlapping clusters with good clustering performance. It was powerful and it could naturally fit into the hybrid military/ commercial communication infrastructure.

Selvi *et al.* [49] worked upon identity based self-delegated signature. A proxy signature scheme was a variant of digital signature scheme in which users delegate their signing rights to another party named as proxy signer. It generates the signature of the actual signer in his absence. They proposed the first identity based self-proxy signature scheme and also introduced a generic scheme. They also defined the appropriate security model for the same and proved both the generic and identity based schemes in the defined security model.

Li *et al*. [50] presented a novel convertible authenticated encryption schemes without using hash functions. An authenticated encryption scheme allows a designated recipient to recover the message and verify its authenticity while keeping the message secret from the public. Hence a convertible authenticated encryption scheme enables the recipient to convert the signature to an ordinary one. It enables the third party to verify its validity in which without the cooperation of the signer, the dishonesty of the signer to any third party could be proved by revealing the message and its converted signature. The limitation was if the recipient could not reveal the converted signature, any third party cannot check the validity of the message even though he gets the message.

Ariffin *et al.* [51] proposed various immune systems approaches for cryptographic algorithm. They also worked on immune-inspired approaches in designing a new function for cryptographic algorithm named as 3D-AES. The immune systems approaches were selected on the basis of complex features which were desirable for substitution and permutation process, that were used to ensure adequate security and confidentiality of the systems in communication model. They identified the correspondences and highlighted the essential computation elements which were applied in cryptographic algorithm to satisfy the Shannon's confusion and diffusion properties. The randomness of the output in the 3D-AES algorithm was comparable with AES algorithm.

Nithyanandam *et al.* [52] discussed recent trends in secure personal authentication for iris recognition using novel cryptographic algorithmic techniques. In biometrics, human being needs to be identified based on some characteristic physiological parameters. A wide variety of recognition schemes are used to confirm the identity of an individual requesting their services. They presented various approaches to generate a unique and more secure cryptographic key from iris template. Biometric cryptosystems had been introduced as a reliable way of concealing private keys by using biometric data. A fuzzy vault refers to a biometric cryptosystem that can be used to effectively protect private keys and to release them only when legitimate users enter their biometric data. Distance metric such as hamming distance is used for the template matching identification process. They showed that the Reed-Solomon error-correcting algorithm can work effectively compare to other techniques.

Verma *et al.* [53] proposed an efficient symmetric key cryptography algorithm for information security. This block encryption algorithm was much faster and offers the enhanced security features compared to other symmetric key algorithms. It helps in achieving confidentiality as well as message authentication. It produces better performance than other common encryption algorithms used in terms of time consumption, whenever there is a change in packet size. It was also good whenever there was a change in data type such as image, audio or video instead of text. By changing key size it also proved that higher key size leads to change in the battery and time consumption.

Ju [54] proposed a lightweight key establishment in wireless sensor network based on elliptic curve cryptography. This protocol combined Elliptic Curve Diffie-Hellman (ECDH) with symmetric cryptography and hash-chain which was used to solve compromise threat and problem of initial key detection. It exhibited less computation complexity, communication cost and storage requirement. It supported different size of sensor networks and flexible against the increase of the network model.

Teerakanok and Kamolphiwong [55] worked upon accelerating asymmetric-key cryptography using parallel-key cryptographic algorithm. They also proposed a new mechanism called Parallel-key Cryptographic Algorithm (PCA) which accelerates the cryptographic system in encryption and decryption process. It strengthens the system against brute force attack. They also showed that the time used by Brute force attack on

PCA was longer than RSA. This algorithm was more flexible for sending or transferring data through insecure channels without any problems of key agreements.

Halkidis *et al.* [56] discussed architectural risk analysis of software systems based on security pattern. They also proposed a methodology for quantifying the security level based on the missing security patterns. Since the estimation could be performed already at the design phase, security problems could be detected at an early stage, which reduced the cost compared to the introduction of security during implementation. The whole process was automated using a methodology which extracts the risk of a software system. It was done by reading the class diagram of the system which was under consideration.

Gupta *et al.* [57] proposed a modern cryptography algorithm used to improved data security. The proposed algorithm was better in terms of speed compared with the existing encryption algorithm; therefore it improves data security by inserting the symmetric layer. It was based on block cipher hence it takes less time if the file size was large. Author claimed that it was almost impossible to break the encryption algorithm without knowing the exact key. This method could be applied for data encryption and decryption in any type of public application for sending confidential data. The main limitation of this work was used assumption was not practically feasible.

Hoch and Shamir [58] proposed a fault analysis of stream ciphers. It could be used to attack the standard constructions of stream ciphers based on Linear Feedback Shift Register (LFSR's), as well as more specialized techniques which can be used against specific stream ciphers. While most of the schemes could be successfully attacked, they pointed out several interesting open problems such as an attack on FSM filtered constructions and the analysis of high Hamming weight faults in LFSR's.

Koo *et al.* [59] worked on implementation and analysis of new lightweight cryptographic algorithm suitable for WSN. Sensor devices had critical resource constraints such as processing speed, memory size and energy supply. They also proved that energy consumption affects the network lifetime so that energy efficiency was an important requirement for WSN. They also analyzed the performance between a new lightweight cryptographic algorithm HIGHT and implemented cryptographic algorithms on TinySec. They concluded that HIGHT was recommended for TinySec as like traditional cryptographic algorithms on TinySec.

He *et al.* [60] proposed a Storage Performance Evaluation Kernel (SPEK) module for block-level storage systems under faulty conditions. SPEK based on direct attached storage and block level networked storage systems. Each SPEK consists of a controller, several workers, one or multiple probers and several fault injection modules. It runs at kernel level and eliminates skews and overheads caused by file systems. It allows a storage architect to generate configurable workloads to a system under test and to inject different faults into various system components such as network devices.

Du *et al*. [61] proposed routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. This model was used for better performance and security. It establishes shared keys for neighbor sensors which communicate with each other. They utilized elliptic curve cryptography in the design of an efficient key management scheme for sensor nodes. The performance evaluation and security analysis process that the key management scheme provides better security with less overhead.

Dongyang *et al.* [62] worked on key management scheme for segment-based document in which encryption keys was assigned arbitrarily. It had some advantages; a) Key generation and derivation was simple and efficient because only hash functions and a rapid symmetric encryption algorithm were used, b) Based on the security of hash and symmetric encryption functions it was secure against collaborative attack, reverse attack and key modification attack., c) Storage space requirements were relatively reasonable. The scheme was suitable for XML-based document format used in web publishing named as Common E-Document Blending XML CEBX.

Jailin *et al.* [63] worked upon performance analysis of hybrid cryptography for secured data aggregation in WSN. They were constrained in terms of computational and energy resources. Simulation processes of the proposed model based upon hybrid cryptography is named as Dynamically Secured Authenticate and Aggregation Scheme (DSAA). It was used to authenticate a person in an indoor environment. The advantages of this method were a) the keys were generated after deployment of the sensors, b) no pre-assignments of keys were required, and c) there was no need to store a look-up key table on the nodes. It also ensured confidentiality, authentication and integrity. The proposed method compares the message digest at the node level itself, which reduced the transmission overheads. Author could not provide the solution of power conservation.

Li *et al.* [64] discussed the application of hybrid encryption algorithm in software security. The traditional information security algorithms were using a single encryption algorithm technology while ignoring its shortcomings. They designed the hybrid encryption algorithm for the straightforward use of well-known encryption algorithm, which improved the well-known encryption algorithms and defined a new initialization encryption algorithm in order to achieve the design of hybrid encryption algorithm They also illustrated the hybrid encryption algorithm in real-life application to prove that the algorithm was practical value.

Wang and Han [65] discussed a provably secured threshold ring signature scheme in certificateless cryptography. It was a group-oriented signature scheme which allowed a member of a group to sign messages on behalf of the group without revealing his/her identity. They also introduced the concept of threshold ring signature into certificateless public key cryptography and propose a concrete certificateless threshold ring signature (CLT-Ring) scheme. The security models of certificateless threshold ring signature were also formalized and secured in the random oracle model.

Bharadwaj and Chakraverty [66] proposed a Circular Design Pattern (CDP) which could support the creation of adaptable, application-specific encryption techniques with varying levels of security and served as a template for generating Dynamic Symmetric Encryption Frameworks (DSEF). They also proposed a modified version of conventional cryptosystem model to deploy the DSEF. CDP exploits reusability maximally to develop new encryption strategies using existing encryption algorithms and DSEF. The limitation was that due to gradual increase of complexity it was maintainable to some fixed pattern.

Huawei *et al.* [67] proposed two modularized, non-interactive converters for designing applied cryptographic protocols which were used to transform secure protocols in the ideal model into those secure in the real model. It was working to create secured and applied protocols in practice because the converters were modular and secured. The converters had the following advantages; a) they are non-interactive, therefore could improve the operating efficiency of the transformed protocols, b) Transformation algorithms were in favor of realizing automatic transformation.

NIU [68] worked upon excellent periodic binary sequence with genetic algorithm. Cryptographic sequences with high linear complexity and high error linear complexity are called excellent sequences. They also designed a genetic algorithm to generate an

excellent periodic binary sequence. They observed that excellent sequences had a highest error linear complexity related to the period. The limitation was that it works only with fast algorithm for determining the linear complexity

Gaps in study and observation

Based on the literature survey following observations have been drawn, stated as:

1) Short keys are not capable to provide adequate security.

2) Algorithm needs complex adaptability to surpass crypto attack.

3) In order to keep all the primitives in limit optimized hardware is required.

4) Use of dynamic keys is preferred for encryption process.

5) Optimized key management is required to achieve more secured cryptosystem.

6) Key size should be as large as to produce infeasibility to an attack.

Objectives

From the Observations drawn from literature survey; following objectives have been derived.

1) To study various cryptographic algorithm.

2) To design an optimized key management scheme for data encryption.

3) Comparison of our approach with existing ones.

4) A technique needs to be theoretically strong and also practically viable.

CHAPTER 3

SYMMETRIC KEY CRYPTOSYSTEM

This chapter provides brief description of symmetric cryptographic algorithms. On the basis of literature survey, some of the needed advancements are also discussed. A new symmetric encryption approach is suggested in this chapter and also fulfilling the gaps stated in previous chapter. At the end MATLAB code is also given.

Symmetric key encryption uses the same key for encryption and decryption of message. The necessity with a symmetric cryptosystem is to transfer secret key to both communication parties before secure communication can begin. The establishment of a shared secret key between communication parties had always been a difficult problem because the task needed a secure confidential channel. The basis of cryptographic systems is to make the plaintext to ciphertext mapping as random looking as we can. Sometimes, using an attack by approximation, we can detect a structural relationship in the operation of an algorithm, which exists with some small bias (i.e., an apparent deviation from random-ness).

Various soft computing techniques can be used in cryptographic algorithm to provide a better secrecy. These techniques are different from traditional algorithms, it uses biological resembles for authenticity of accessing user. These techniques can also detect the compromised states or algorithms attempted to attack by adversaries. Soft computing techniques resemble biological processes which are largely based on format logical systems. Intrusion Detection System (IDS) inspects all inbound and outbound activity and identifies suspicious pattern that may identify a system attack. It basically uses two algorithms; 1) Error Back Propagation (EBP), is a training algorithm used for feed forward artificial neural networks. 2) Radial Basis Function (RBF), is a neural network which is based on supervised learning. It is concluded that RBF is better than Error Back Propagation.

Some schemes are based on the assumed computational difficulty of solving instances of particular combinatorial problems. Solving such problems is not that much complex as it looks. Cryptography is based on the idea that factorisation really is hard. Symmetric key encryption scheme which are of most used in present security systems are as:

3.1 Data Encryption Standard (DES)

DES is symmetric block cipher algorithm. It encrypts 64 bits block at a time and produces the output of same block length. It uses 56 bit as a key. DES actually accepts a 64 bit key; the remaining eight bits are used for parity checking and have no effect on its security.

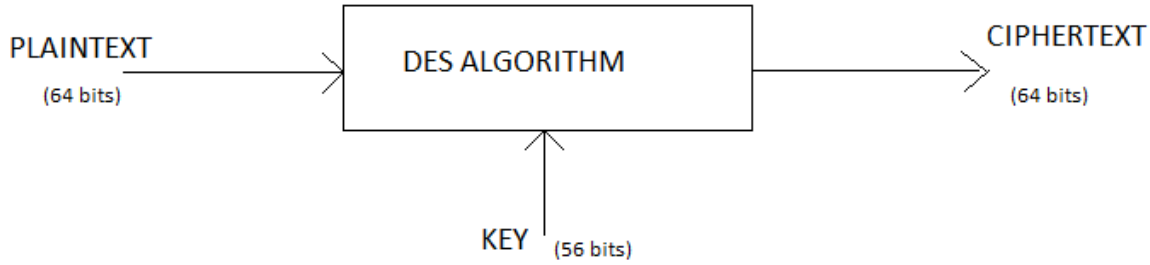


Figure 3.1: Data Encryption Algorithm

Initially, plaintext to be encrypted is divided in blocks of 64 bits; if number of bits in the data is not divisible by 64 then last block will be padded. It performs substitution and permutation a number of times in iterations called rounds. Non-linearity introduced into encryption with the use of S-boxes to make decryption infeasible to adversaries.

DES performs initial permutation on 64 bit block of data. Then it is split in two parts Li and Ri of 32 bits each. Both parts passed through a round where Ri is permuted with expansion table and ex-ORed with 48 bit sub-key which is parallel derived from 56 bits input key. Then it is passed through S-boxes and P-boxes. After permutation it is ex-ORed with Li, results as Ri+1; and Li+1 = Ri, which become input for second round. For each round a different 48 bit sub key is provided.

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

Table 3.1: Initial permutation

By passing through 16 rounds of these substitution, permutation and ex-OR, it is send to final permutation which is basically inverse of initial permutation. The output of final permutation is ciphertext. DES exhibits a strong avalanche effect which strengthens the security.

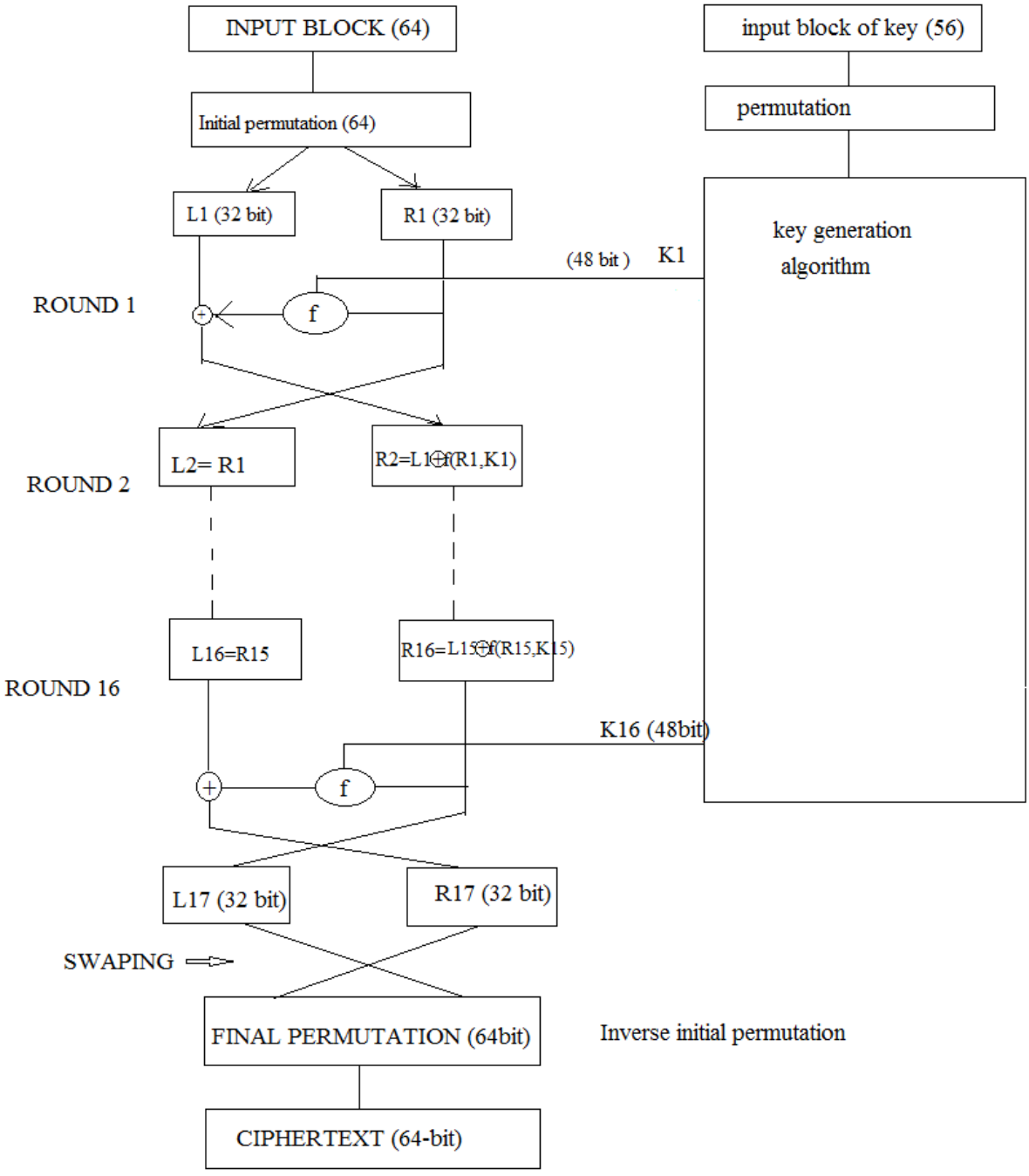


Figure 3.2: DES Encryption Algorithm

3.1.1 DES Subkey Generation

For Sub key generation from input key, 56 key bits (after permutation) divided into two 28-bit halves.

- Each half circularly shifted left by one bit (rounds 1, 2, 9 and 16) or 2 bits (all other rounds).
- Halves recombined into 56 bit string. It is again compression permuted which produces 48 bit subkey for corresponding round.

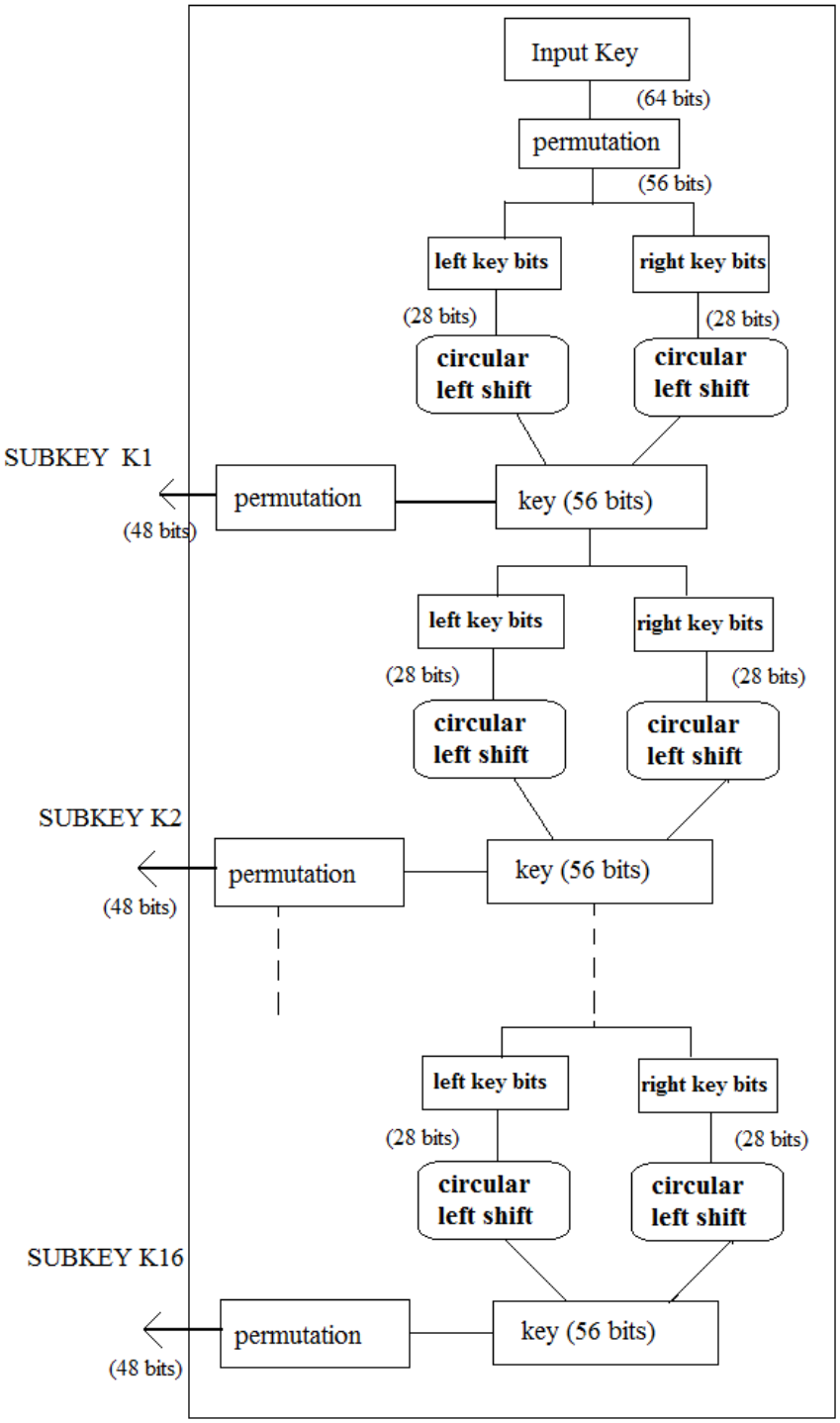


Figure 3.3: Key generation algorithm

- For all rounds of DES, 16 different subkeys are generated through this algorithm.

3.1.2 DES Each Round Working

Input block is divided in two 32 bit halves. Then right block is passed through expansion permutation which produces 48 bit output. It is EX-ORed with corresponding 48 bit sub-key. Then it is passed through S-Boxes which produces 32 bit block.

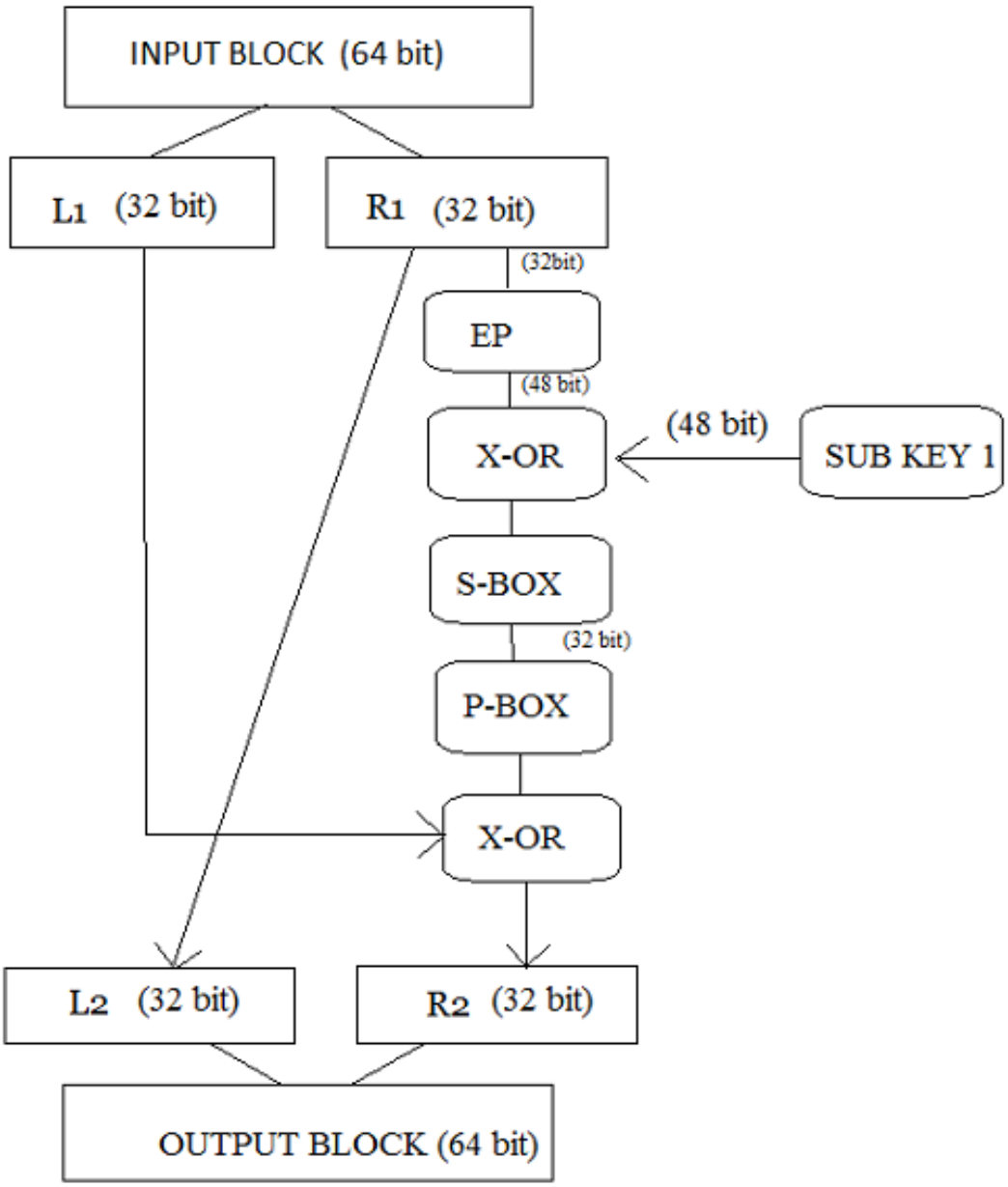


Figure 3.4: DES single round

S-Boxes perform substitutions. There are 8 different S- boxes which map 48 bit input to 32 bit output. These are non-linear tables. It maps 6 bit input as a 4 bit output. Each table consists of 4-row and 16-columns. Input's first and last bit specifies the row and middle 4 bits specifies the column. For ex. S-box1

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

Table 3.2: S-Box 1

If 6-bit input 011010; row 0 and column 13 i.e.9 = 1001 (output 4-bit)

For the Input 110010; row 2 and column 9 i.e.12 = 1100 (output).

The output of S-Boxes is permuted through p-boxes. Now, permuted data is EX-ORed with the left 32 bit of same round. Output becomes right 32 bit of next round. Left 32 bit for next round is taken as it is of right part of present round. Similarly 16 rounds performed. The left and right parts are swapped. This 64 bit is final permuted which is inverse of initial permutation. The output of final permutation is ciphertext.

Decryption process is similar to encryption only the sub keys are operated in reverse order.

DES is simple and fast in operation. The small key size (56 bit) is the biggest limitation of this approach. It is vulnerable to brute force attacks and can be broken. Better secrecy can be achieved as key length increases, resulted as Triple DES encryption algorithm.

3.2 TDES ALGORITHM

TDES is also a symmetric block cipher. It basically operates DES algorithm three times. It takes three 56-bit keys, for an overall key length of 168 bits. The entire 168-bit is typed once instead of entering each of the three keys individually and Triple DES breaks the key into three subkeys, padding can also be done to make each key 64 bits long. The procedure for encryption is the same as regular DES, but it is repeated three times, hence the name Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

Triple DES encrypts input data three times. The three keys are referred to as k1, k2 and k3.

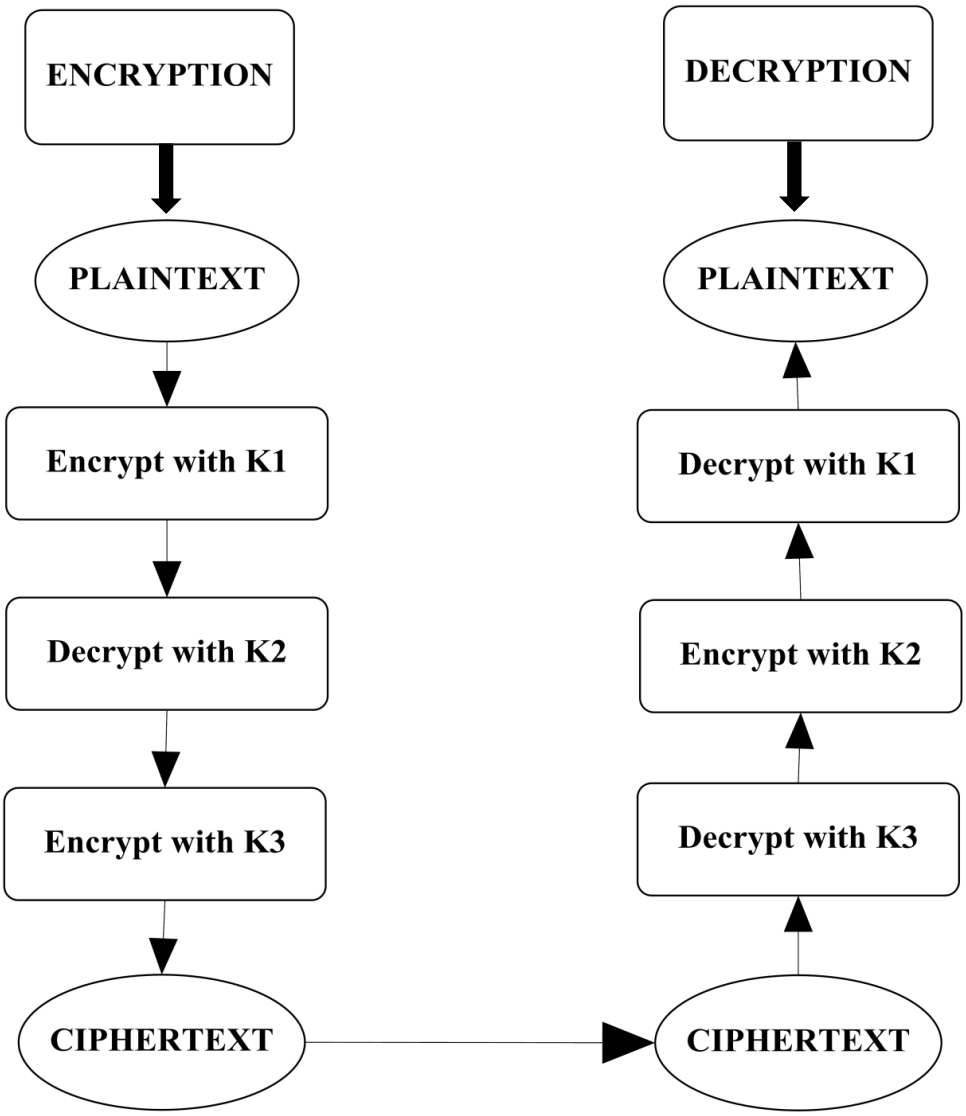


Figure 3.5: TDES Encryption and Decryption

Triple DES provides a relatively simple method of increasing the key size of DES to protect against brute force attacks, without need of adopting a new cipher algorithm. Three keys are referred to as bundle keys with 56 bits per key. There are three keying options in data encryption standards:

- Keying option 1: All three keys are independent.
- Keying option 2: K1 and K2 are independent, and K3 = K1.
- Keying option 3: All three keys are identical, i.e. K1 = K2 = K3.

Keying option 1 provides the strongest encryption, with 3 x 56 = 168 independent key bits.

Keying option 2 provides less security, with 2 x 56 = 112 key bits. This option is stronger than simple DES encrypting twice, e.g. with K1 and K2, because it protects against meet-in-the-middle attacks.

Keying option 3 is similar to DES, with only 56 key bits. This option provides backward compatibility with DES, because the first and second DES operations simply cancel out. Key option 3 is known as triple DES. The triple DES key length contains 168 bits but the key security falls to 112 bits.

Triple DES with three independent keys (keying option 1) has a key length of 168 bits (three 56-bit DES keys), but due to the meet-in-the-middle attack the effective security it provides is only 112 bits. Keying option 2 reduces the key size to 112 bits. It is susceptible to certain chosen-plaintext or known-plaintext attacks.

Triple DES runs three times slower than DES, but is much more secure if used properly. The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse. Like DES, data is encrypted and decrypted in 64-bit chunks. 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 each byte. These parity bits are ignored; only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. This means that the effective key strength for Triple DES is actually 168 bits because each of the three keys contains 8 parity bits that are not used during the encryption process.

3.3 AES Algorithm

AES is a symmetric block cipher. The input block and output block data each are a fixed length size of 128 bits. An input key required as input to the AES algorithm can be of 128, 192 or 256 bits. The same key is used for both encryption and decryption. In general, the longer the key, the higher the security level obtained with the encryption. AES is efficient encryption scheme in term of degree of security and complexity involved in operation.

AES Processing Steps

The AES algorithm consists of a series of steps. After the initial key expansion operation, all the steps are repeated a number of times; each such repetition of steps is called a round. The number of rounds used in the algorithm dependens on the size of the input key. The number of rounds are n=10 for a 128-bit key, 12 for a 192-bit key, and 14 for a 256-bit key.

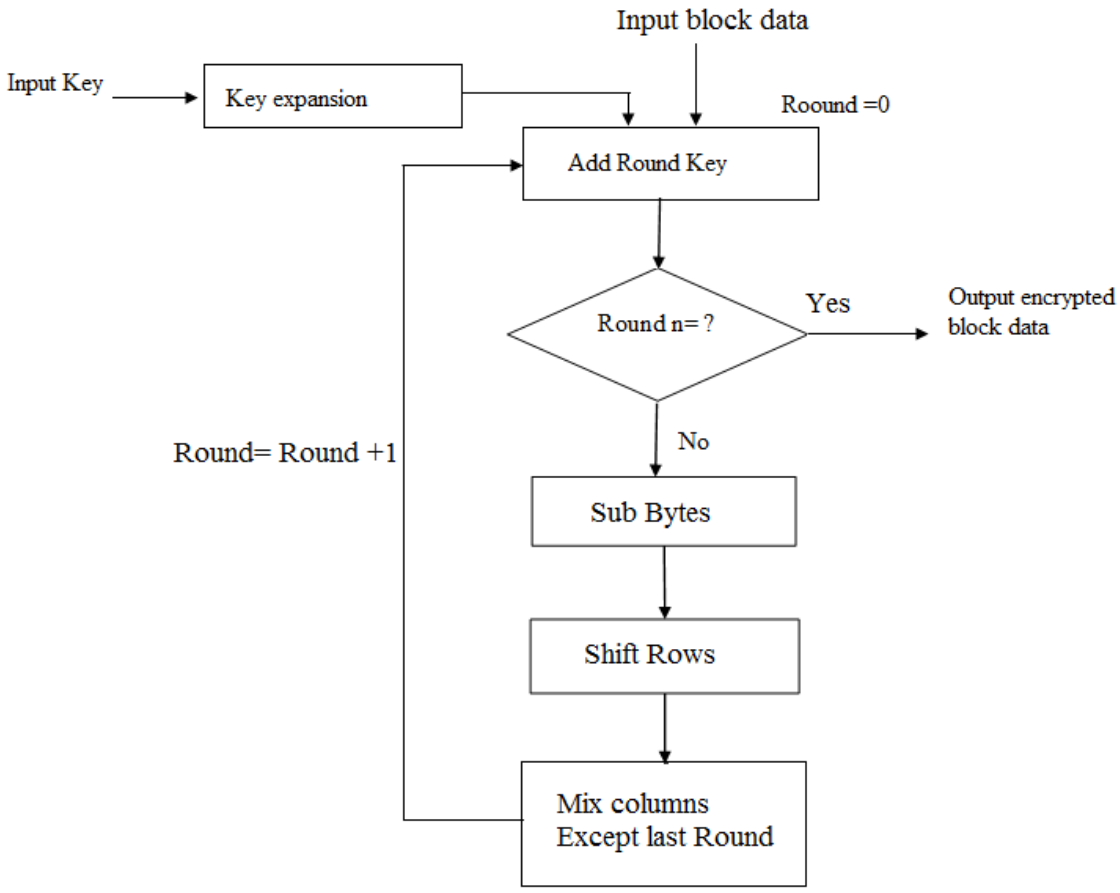


Figure 3.6: AES Encryption Processing Steps

The input plaintext data is first used to create a matrix of 4x4 bytes of data. The transformed data of this matrix is stored as state data during processing of each step.

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Figure 3.7: State Data

The encryption steps are briefly described below:

The **Key Expansion** step expands and transforms a 128-, 192- or 256-bit key to 11, 13, or 15 sub-keys, each 128-bits long, using the Rijndael key expansion algorithm. One sub-key corresponds to each AES processing round, thus each sub-key is referred to as a round key. The set of 11, 13, or 15 round keys comprises the key schedule.

The **Add Round Key** step is a transformation; it combines the current state data block and the round key corresponding to the specific round using an XOR function.

The **Sub Bytes** step replaces each state data byte with an entry in a fixed lookup table.

The **Shift Rows** step rotates the four bytes of state data in each row in the state data matrix.

A0,0	A0,1	A0,2	A0,3
A1,0	A1,1	A1,2	A1,3
A2,0	A2,1	A2,2	A2,3
A3,0	A3,1	A3,2	A3,3

Current State

A0,0	A0,1	A0,2	A0,3
A1,1	A1,2	A1,3	A1,0
A2,2	A2,3	A2,0	A2,1
A3,3	A3,0	A3,1	A3,2

Next State

Figure 3.8: AES step Shift Rows

The **Mix Columns** step performs a transformation on the four bytes of state data in each column in the state data matrix.AES is operated in one of the modes (ECB, CBC, CFB, and OFB) on modes for encryption or decryption to be performed.

Encrypting data with different keys may take different times. The key and the data affect the time taken to encrypt or decrypt, which gives a leakage of information about the key. Even monitoring power consumption may reveal which instructions are being executed.

The above discussed symmetric key schemes can be compared using various parameters. It gives the significance of each scheme, shown in table below:

Table 3.3: Comparison of DES, TDES and AES

PARAMETER	DES	TDES	AES
Key size	56 bits	168 bits	128, 192, 256 bits
Block size	64 bits	64 bits	128 bits
Speed	Slow	Very Slow	high
Degree of security	Very low	moderate	Highly secure
Resource consumption	high	moderate	low

3.4 Security Analysis

Different algorithms provide different degrees of security; it depends on how hard they are to break. If the cost required to break an algorithm is greater than the value of the

encrypted data, then the algorithm is supposed to be safe. If the time required breaking an algorithm is longer than the time that the encrypted data must remain secret, and then also it is safe. If the amount of data encrypted with a single key is less than the amount of data necessary to break the algorithm, it is supposed to be safe. Cryptography is based on the idea that factorisation really is hard. Mathematical functions simply map inputs to outputs and exist in some conceptual space, but when we implement a function, the computation consumes resources: time and power. Attackers do it by observing some parameter.

Cryptographic algorithms are usually strong against mathematical attacks but their practical implementations, both in software and hardware opened the door to side-channel attacks [19]. Without expensive equipment or intrusive monitoring, these attacks bypass the mathematical complexity and find the cryptographic key by observing the power consumption or the execution time variations of the device in normal operation mode.

It has been long assumed that physically generated random noise can yield cryptographically unpredictable random numbers. These preliminary designs are the first step in a several-step process of producing cryptographic random numbers using system on chip devices.

Online commerce activities, transactions and services are possible if communications over open networks can be conducted in a secure manner; hence the need of the hour is an efficient but simple, encryption and decryption algorithm for encrypting any kind of data. Also, the keys required for encryption and decryption must be randomized and randomly generated from the changing data bits. My work tries to fulfil this to some extent. In this work, we explore techniques to improve the performance of symmetric key cipher algorithms.

3.5 Proposed Encryption Scheme

A. Our new proposed algorithm has a two-level algorithm which consists of:

a) Parity checking level

b) Dibit checking level

B. Proposed scheme can be understood easily by the following case study. Let the 8-bit data be 10 11 01 10.

Encryption algorithm

Step I) The 8 bits are indexed at the very outset for easy reference as shown in table 1.

Step II) Parity checking level starts encryption here in this algorithm. The fundamental aspect of this level is that the key for encryption is stored in the data bits pattern itself and

this key varies according to the varying data bits. The encryption is done in bit-by-bit operation. For encrypting data bits having index number 1 3 5 7, its corresponding key is data bits having index number 2 4 6 8. The two nibbles are then checked bit-by-bit so that the final result has even parity with data bits 1 3 5 7. This final result is the pseudo-encrypted bit pattern 1` 3` 5` 7` as shown in table 2. The pseudo-encrypted bit pattern 1` 3` 5` 7` becomes the key for the bit pattern 2 4 6 8. In the same manner, even parity is checked with the key and we get the pseudo encrypted bit pattern 2` 4` 6` 8` shown in table 3. Let us understand this with our example.

The same parity checking is done for the data bits having index number 2 4 6 8. In this case the key will be the pseudo-encrypted bit pattern 1` 3` 5` 7`. Thus the complete pseudo-encrypted bit pattern having index number 1` 2` 3` ….8` becomes: 11 01 10 00.

Step III) The level 2 (di-bit checking level) starts here. In the same way as level 1, the 8-bits are indexed as 1` 2` 3`…8` for easy reference (Table 4).

Step IV) Bits having index numbers 1` and 2` are extracted from the 1 byte data. The dibit pair 1` 2` (consisting of the bits having index number 1` and 2`) becomes the reference dibit for the other three dibit pairs (3`-4`, 5`-6` and 7`-8`). Now a virtual table is prepared with four columns. The fields in the columns are NC (not complement), LSB-C (least significant bit-complement), MSB-C (most significant bit-complement), FC (full complement). In order to represent the four fields, we need at least 4 bits. Let the four fields, NC, LSB-C, MSB-C and FC be represented respectively as 00, 01, 10 and 11. Our objective in this step is to obtain the dibit pairs 3`-4`, 5`-6` and 7`-8` from the reference bit 1`-2` by any one of the four methods NC, LSB-C, MSB-C or FC; NC means the dibit pair is same as the reference dibit; LSB-C means the dibit pair is obtained from the reference dibit pair by complementing the least significant bit, MSB-C means the dibit pair is obtained from the reference dibit pair by complementing the most significant bit and FC means the dibit pair is obtained from the reference dibit pair by complementing both the bits. The logic of the encryption then follows the following rule: Dibits 3`-4`, 5`-6` and 7`-8` are coded with their individual field codes and the bit 1` and 2` are appended both at the beginning and at the end respectively.

INDEX	1	2	3	4	5	6	7	8
BITS	1	0	1	1	0	1	1	0

Table 1: Indexing of bits

1	1	0	1	// data bits 1 3 5 7
0	1	1	0	// data bits 2 4 6 8 (key)
1	0	1	1	// pseudo–encrypted bits 1’ 3’ 5’ 7’

Table 2: Encryption of bits 1 3 5 7

0	1	1	0	// data bits 2 4 6 8
1	0	1	1	// data bits 1’ 3’ 5’ 7’ (key)
1	1	0	1	// pseudo-encrypted bits 2’ 4’ 6’ 8’

Table 3: Encryption of bits 2 4 6 8

INDEX	1’	2’	3’	4’	5’	6’	7’	8’
BITS	1	1	0	1	1	0	1	1

Table 4: Indexing of bits

REFERENCE DIBIT 11	DIBIT PAIR	00 NC	01 LSB-C	10 MSB-C	11 FC
	3’-4`			▲	
	5’-6`		▲		
	7’-8`	▲			

Table 5: Formation of virtual table

Let us understand this with an example.

The dibit pair 1`-2` (11) is extracted. This becomes the reference dibit for the other three dibit pairs 3`-4` (01), 5`-6` (10) and 7`-8` (11). Now the logic goes like this: 01 (Dibit pair 3`-4`) is obtained from 11 (reference bit) by the method MSB-C; 10 (Dibit pair 5`-6`) is obtained from 11(reference bit) by the method LSB-C; 11 (Dibit pair 7`-8`) is obtained from 11 (reference bit) by the method MSB-C.

The virtual table looks as shown in table 5. NC corresponds to the code 00, LSB-C corresponds to the code 01, MSB-C corresponds to the code 10 and FC corresponds to the code 11. Thus, the dibit encryption follows as:

Dibit pair 3`-4` is coded as 10, dibit pair 5`-6` is coded as 01 dibit pair 7`-8` is coded as 00. And the bits 1` and 2` are appended at the beginning and at the end respectively. Thus the final encrypted code word becomes: 11 10 01 00.

Decryption algorithm

Step I) The 8 bits of encrypted data are indexed again at the very outset for easy reference (Table 6).

Step II) The bits having index numbers 1 and 2 are extracted from the 1 byte data. The dibit pair 1-2 (consisting of the bits having index number 1 and 8) becomes the reference dibit for the other three dibit pairs (3-4, 5-6 and 7-8). The virtual table having the fields NC, MSB-C, LSB-C and FC is prepared again and the codes are entered accordingly in the appropriate places. The table looks like (Table 7).

For dibit 3-4 ▲ is placed in MSB-C field indicating the pseudo-decrypted dibit 3`-4` is obtained by complementing the least significant bit of the reference dibit. For dibit 5-6, ▲ is placed in MSB-C field indicating the pseudo-decrypted dibit 5`-6` is the same as the reference dibit. And for dibit 7-8, ▲ is placed in NC field indicating the pseudo-decrypted dibit 7`-8` is obtained by complementing both the NC of the reference dibit. Thus, we get: 1 11 10 01 00, the pseudo decrypted bit pattern.

Step III) The pseudo-decrypted bit pattern goes for the next stage of decryption logic. For decrypting data bits having index number 2` 4` 6` 8`, its corresponding key is data bits having index number 1` 3` 5` 7`. The two nibbles are then checked bit-by-bit so that the final result has even parity with data bits 1` 3` 5` 7`. This final result is the decrypted bit pattern 1 3 5 7. The decrypted bit pattern 1 3 5 7 becomes the key for the bit pattern 2` 4` 6` 8`. In the same manner, even parity is checked with the key and we get the decrypted bit pattern 2 4 6 8 (Table 8). The same parity checking is done for the data bits having index number 1` 3` 5` 7`. The key in this case is the decrypted bit pattern 2 4 6 8 (Table 9). Thus the complete decrypted bit pattern having index number 1 2 3 ….8 becomes: 10 11 01 10. This is exactly the same as the original plaintext 10 11 01 10.

INDEX	1	2	3	4	5	6	7	8
BITS	1	1	1	0	0	1	0	0

Table 6: Indexing of bits

REFERENCE DIBIT 11	00 NC	01 LSB-C	10 MSB-C	11 FC
				
				
				

Table 7: Formation of virtual table

1	1	0	1	// data bits 2’ 4’ 6’ 8’
1	0	1	1	// data bits 1’ 3’ 5’ 7’ (key)
0	1	1	0	// decrypted bits 2 4 6 8

Table 8: Decryption of bits 2 4 6 8

1	0	1	1	// data bits 1’ 3’ 5’ 7’
0	1	1	0	// decrypted data bits 2 4 6 8 (key)
1	1	0	1	// decrypted bits 2 4 6 8

Table 9: Decryption of bits 1 3 5 7

MATLAB CODE:

Its simulation has been successfully implemented in MATLAB. Its complete hardware implementation can be performed in Verilog HDL also.

```
clear all;
clc;
a=input('enter the data byte ','s');
b=bin2dec(a);
bit_array=bitget(b,8:-1:1);
for i=1:2:7
ps_enc(i)=xor((bit_array(i)),(bit_array(i+1)));
end
for i=2:2:8
ps_enc(i)=xor((bit_array(i)),ps_enc(i-1));
end
display('The pseudo encrypted data byte is '); display(ps_enc);
```

```
%preparing the reference bit
ref_bit(1)=ps_enc(1);
 ref_bit(2)=ps_enc(2);
display ('The reference bit is ');ref_bit
%the final encryption algorithm
for j=3:2:7
final_enc(j)=xor((ref_bit(2)),(ps_enc(j)));
final_enc(j+1)=xor((ref_bit(1)),(ps_enc(j+1)));
end
final_enc(1)=ref_bit(1);
final_enc(2)=ref_bit(2);
display ('The final encrypted data byte is ');final_enc
%preparing the decryption algorithm
for k=3:2:7
ps_dec(k)=xor((ref_bit(2)), (final_enc(k)));
ps_dec(k+1)=xor((ref_bit(1)), (final_enc(k+1)));
end
ps_dec(1)=ref_bit(1);
ps_dec(2)=ref_bit(2);
display('The pseudo decrypted data byte is ');ps_dec
for m=2:2:8
final_dec(m)=xor((ps_dec(m)),(ps_dec(m-1)));
end
for m=1:2:7
final_dec(m)=xor((ps_dec(m)),(final_dec(m+1)));
end
```

display('The final decrypted data byte is ');final_dec.

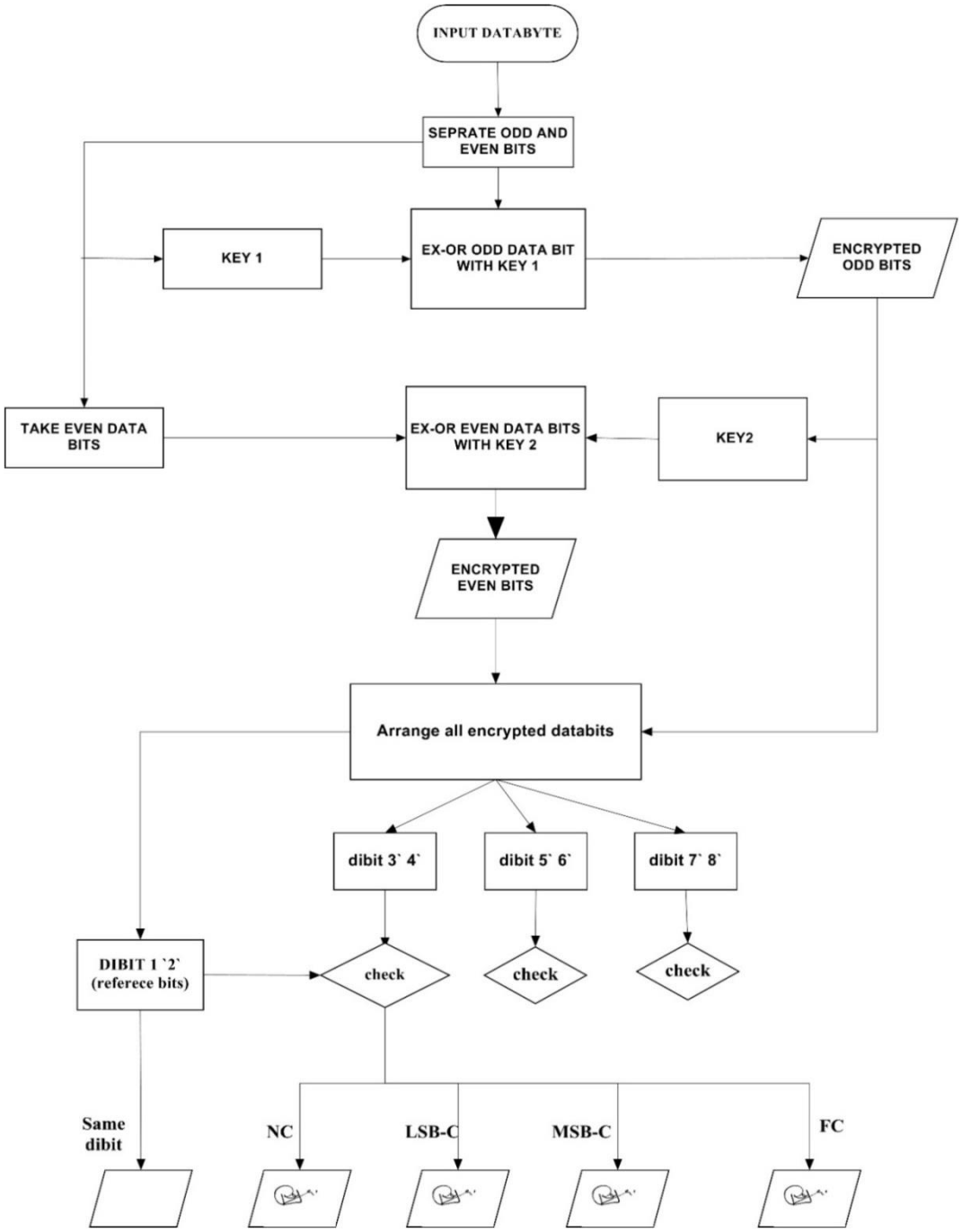


Figure 3.9: Flow-chart Encryption scheme.

3.6 KEY MANAGEMENT SYSTEM

Key Management System (KMS) refers the techniques for controlling the distribution [39], use and update of cryptographic keys. It supports updating of keying material, key backup/recovery, revocation [42] and managing certificates in certificate-based systems.

Definition: KMS is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties. A keying relationship is the state wherein communicating entities share common data (keying material) to facilitate cryptographic techniques. This data may include public or secret keys, initialization values and additional parameters.

KMS should be able to generate a secret key between two parties. It aims to store key, to prove the authenticity of the keys and of the communicating parties. It has a high priority for longer security of cryptographic applications. As soon as a key has been generated it must be prevented from the reach of third parties.

The first component of a typical key management system [13, 47] is building a database which is used for storing the key. There are several key classes like public, private and session keys. For each class of keys different way of storage is needed. Key generation is the second component of a key management system. A reliable session key has to be implemented to provide an effective encryption at the data transfer. To find a good key at generating session key is necessary, if the good session key cannot be generated; there is no perfect key exchange method for it. Key management [48, 61] is basically a bundle of techniques and procedures supporting:

1. Initialization of system users within a domain;
2. Generation, distribution, and installation of keying material;
3. Controlling the use of keying material;
4. Update, Revocation, and destruction of keying material; and
5. Storage, Backup/Recovery, and archival of keying material.

A proper key management includes generation of keys, injection of keys in the devices, introducing keys (symmetric secret key or asymmetric public key) in the database, and renewing keys for devices in the field. For asymmetric cryptography, a public key infrastructure (PKI): a system to create, manage, distribute, and revoke digital certificates is established.

Key Storage and Protection

In both symmetric and asymmetric cryptography, keying material must be protected from exposure. Key management system enables storage of the key so that it is available for use but still kept confidential. The fields in the key database are encrypted internally before being written to disk. Keys retrieved with end users may need to be stored locally.

Stored symmetric keys and asymmetric private keys must also be protected from exposure. For asymmetric public keys, storage for retrieval and protection from modification is necessary.

Key Retrieval from Local Storage

Keys are identified by name. Several keys associated with one name, varying by algorithm, key size are possible. KMS enables retrieval of a key set from the local storage by an identifier or other attribute, with the choice of the appropriate key from the set.

Key Expiration and Storage Policies

Key management enforces expiration based on the information gained when the key is generated or retrieved from an external source. Keys are removed from the storage after a short lifetime; these keys are not deleted from the database if they have been saved. When KMS generates a new key, it is placed in the cache. An additional KMS function must be called to add a key to database. KMS also allow the storage of expired keys; this is important for keys that are used to sign documents in a specified time period. These keys can be considered to have no expiration but were only considered valid during the specified period.

Key Import and Export

KMS provide import and export between a set of common key and certificate formats as necessary. It can export keys in its own database for distribution.

Symmetric Key Distribution

In symmetric cryptography, the symmetric key must be communicated between the peers. In unicast applications, there are only two peers. Key Management System will provide support for communicating the symmetric key while preventing it from disclose.

Managing cryptographic keys is a critical part of the information lifecycle. The main problems associated with cryptographic key management include:

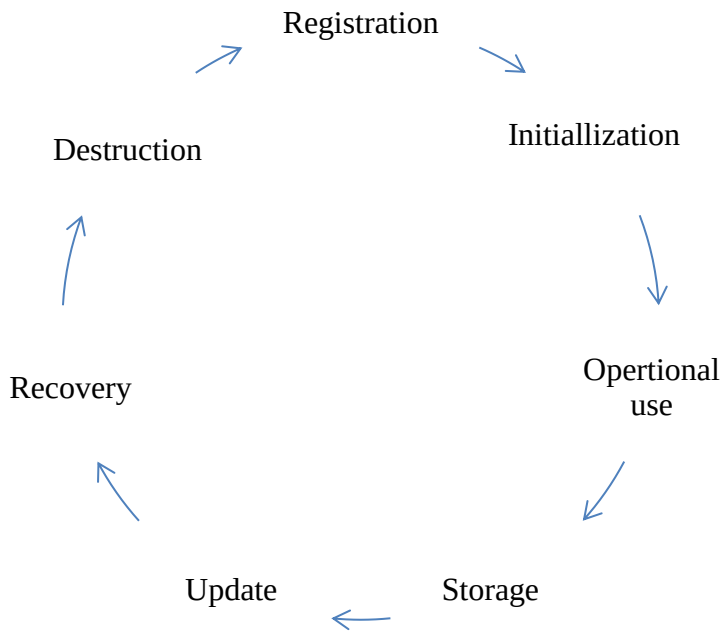
- Realising that crypto, e.g. key and certificates require updating before they expire.
- Keeping track of how to update crypto in legacy systems.
- Locating the devices that require updating and getting to and from the location.
- Following the right procedure for updating system certificates and keys.

Ensuring that the right key is in the right place at the right time is necessity. It is a complicated requirement as an ever-increasing number of keys, while reducing the risk of

internal and external adversary, keeping costs at a minimum. Keys can be securely generated and pushed to any key distribution target as and when required, while vastly improving workflows [54].

Key Management Lifecycle

Cryptographic key management system encompasses the entire lifecycle of cryptographic keys and other keying material. A key has several states during its life, some of them are stated as:



User Registration

During registration, an entity becomes an authorized member of a security domain. It includes the creation and exchange of initial keying material.

Initialization

System initialization: setting up/configuring a system for secure operation.

User initialization: an entity initializes its cryptographic application.

Keying Material Installation

Keying material is installed for operational use, when the software, hardware and crypto module is initially set up, or when new keying material is added to the existing keying material, when existing keying material is replaced then also. Test keying material must be replaced prior to operational use.

Key Registration

Keying material is bound to information associated with a particular entity. Authorization information is performed when the entity is a participant in a key management infrastructure.

Operational Use

The objective is to facilitate the operational availability of keying material for standard cryptographic purposes. A key remains in operation until the end of the key's period or till it is manually not altered.

Storage of Keying Material

It enables confidentiality, integrity, long term availability association. It stores as: a) Operational Storage, b) Backup Storage, c) Key Archive Storage. Storage of keying material depends on type, protection requirements, and stage. When required for operational use, and not present in active memory, acquired from operational storage. If in active memory, or operational storage is lost or corrupted, may be recovered from backup storage and also from archival storage. It shall be stored for immediate availability to an application.

Recovery: It enable user to avail key in case of key loss. It is possible from key escrow with a trusted third party.

Destruction: Using key revocation, compromised keys are removed from system. It prevents the misuse of further information encrypted with the hacked algorithm or system. Finally, Key management is the basis for authentication and secure communication [61-62].

OUTCOME

Symmetric cryptosystem have been discussed. A scheme has been proposed for optimization of symmetric key algorithms. It is generating the key according to data bits. The work has been carried out using MATLAB. Key Management System has also been discussed for better secrecy.

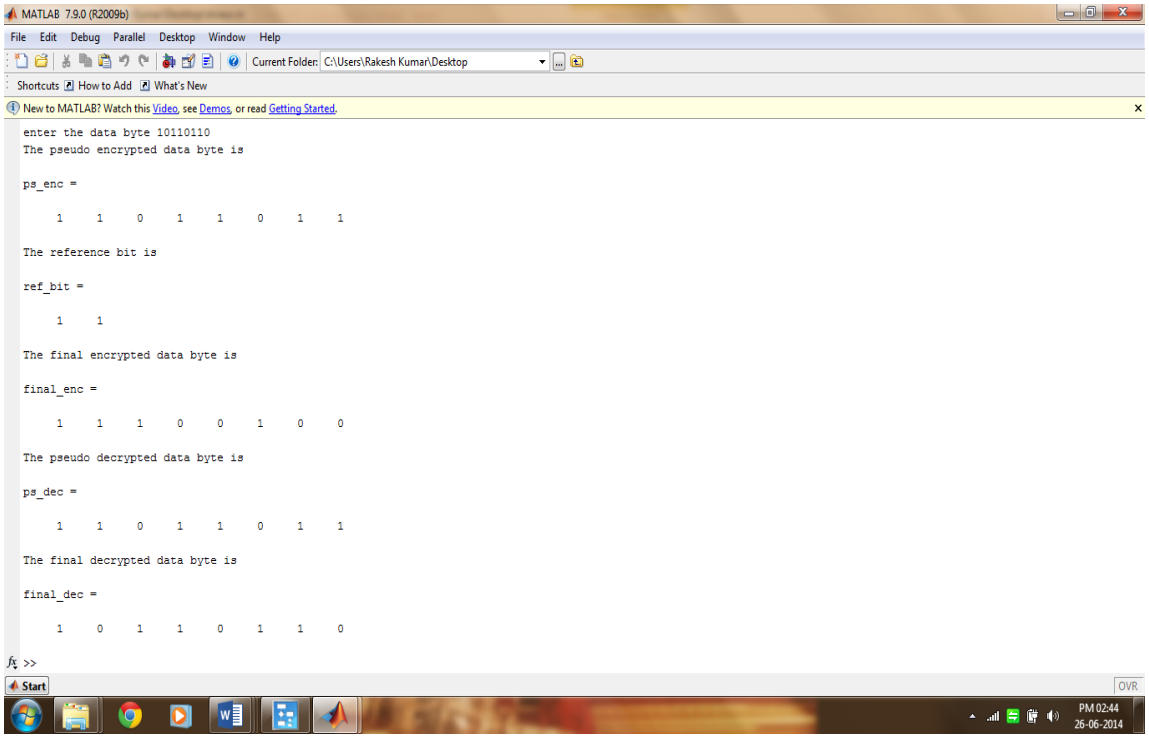
CHAPTER 4
RESULTS AND DISCUSSION

This chapter shows the results with encryption scheme proposed in chapter 3. Discussion on the scheme and its output has been presented. Finally, the comparison of our proposed scheme with existing algorithm has also been shown in this chapter.

4.1 Simulation Detail

The symmetric encryption approach discussed in chapter 3 can be successfully implemented in MATLAB. Since MATLAB is an excellent platform for interfacing and modification is easy so its .m file introduced here. Using any binary data, results have been shown below:

4.2 Results



4.3 Discussion

Results shows that final encrypted data is twice scrambled, first bits in plaintext is arranged in sequence then by ex-oring the odd bits with even number of bits in an order, its output bits are used as key bits for remaining half and both ex-or resultant bits are written in a table. After first scrambling the two reference bits (first and last bit) called dibit is extracted. Then for second time encryption, it uses another table where first encrypted data bits arranged according to its algorithm. Now, the corresponding bit which

is derived from table is communicated to another party. Here another party already knows the encryption scheme it used. With the knowledge of that reference table, communicating party first derived the scrambled data. Then again performing ex-or operation on alternate bits the original data is recovered, which is nothing but plaintext. If the plaintext message input to a basic cryptographic function has a random distribution, then the function provides a strong protection in hiding the plaintext information, even down to the level of an individual bit.

In our approach, key derived from the data bits itself so key used to encrypt is different each time and it has no correspondence with the previous used key. It makes infeasible to attacker to map the relation between plaintext and ciphertext. With a small change in data, ciphertext is completely changed; it is highly required to remain the safe our data. With the symmetric crypto system the major problem is distribution of secret key and to prevent it from untrusted party. In our approach the problem of key distribution is also removed as it is being generated by data itself. As cryptography should be readily available to everyone, simplicity and ease to implement make this scheme most usable approach for small size application in present scenario.

4.4 COMPARISON

Symmetric Key Encryptions are different with respect to key length, input data size and algorithms used to encrypt; each have their own advantages and complexity involved. Execution time is the time taken by processor to encrypt and decrypt message. It also affects the efficiency of encryption scheme. The comparison of proposed scheme with existing algorithm shown below:

Table 4.1: Execution time for different data size

Input (kb)	Execution Time (msec)			
	DES	TDES	AES	Proposed
49	33	48	38	24.5
59	29	54	56	29
100	49	81	90	50
247	47	111	112	120
321	82	167	164	159
694	144	226	210	347

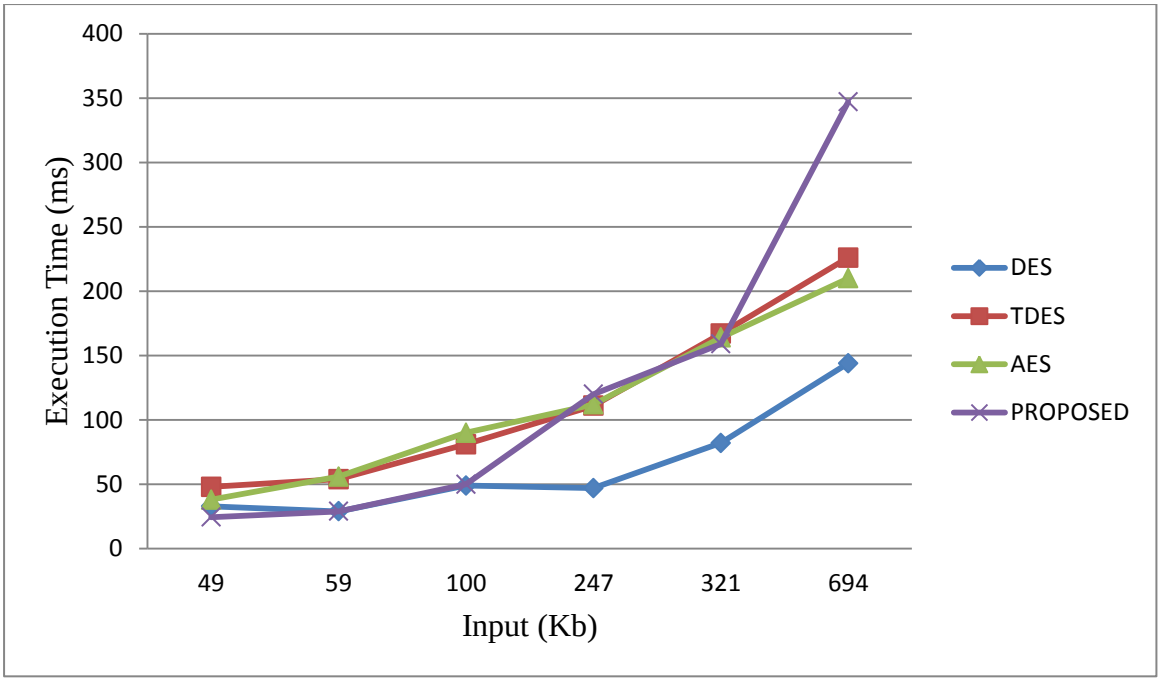


Figure 4.1 Execution time for different data sizes

The graph indicates that proposed approach gives the better results compare to others. This approach is highly suitable for short messages seeking low complexity with adequate degree of encryption security. This Encryption schemes is appropriate for both database (stored data) and communication information (data in transaction).

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

This chapter is dedicated to the summarization of the results presented in the previous chapter, as well as the conclusions that can be derived from the gathered and collated data. Recommendations for actions as well as further studies are also included in this chapter.

Cryptographic algorithms are tools to provide a secure communication. To protect the confidential data from hackers, dependence on cryptography is increasing more rapidly.

This thesis deals with the key management techniques. We examined, the basic requirement with all security algorithm is randomness in original data which comes from the optimized key generation and its management. In most of cryptosystems the security relies on secrecy of key than algorithm. Longer key provide better security against brute force attack but it increases the data volume in ciphertext.

An important edge of our new algorithm is that the keys are generated automatically once the data bits come to the user for encryption. And every time the keys are generated it is random and has no correspondence with the previously generated keys.

In chapter 1 we have given a brief introduction of security goals, which shows the necessity of cryptographic algorithms in present scenario. Classification of cryptography is shown in introduction part. Advantages and disadvantages of both types of cryptography discussed. Key management system in different classes of cryptographic techniques has also been presented. Significance of key length is also presented at the end of chapter 1.

In chapter 2 we presented work done by various researchers in cryptography. We examined basic requirements of cryptographic algorithm used in database. We also investigated several possible techniques used to fulfill security goals. Observations and gaps have also been derived from literature survey in this chapter.

In chapter 3 we represented the symmetric key cryptosystems. Study of DES, TDES and AES algorithms briefly presented in this chapter. We have also suggested a new symmetric approach for encryption purpose.

In chapter 4 we have given simulation details of encryption approach introduced in chapter 3. Results in MATLAB have been pictured in this chapter. Results and discussion shows, used approach mitigate the danger of exposing of private data. Discussion also proves the reliability and simplicity of our approach with adequate cryptographic fulfilment.

Future scope

Symmetric cryptology is an old but still a strong security mechanism that is most widely used. Proposed work can be further extended if randomized padding schemes are used for strong bit security level.

More secured results can be obtained if the key size is further increased but it also increases the processing time. So, there is a trade-off between security level and processing time.

PUBLICATIONS

[1] Rakesh Kumar and Ajay Kakkar, "A review paper on Cryptographic Algorithms for Data Security and Significance of key length," *International Conference on Emerging Technologies in Electronics & Communication*, GNDU Amritsar, pp. 203-205, 2013.

REFERENCES

[1] M. Z. H. Sarker and M. S. Parvez, "A Cost Effective Symmetric Key Cryptographic Algorithm for Small Amount of Data," *IEEE International Multitopic Conference*, pp. 1-6, 2005.

[2] A. D. Keromytis, J. L. Wright and T. Raadt, "The Design of the Open BSD Cryptographic Framework," 2003.

[3] H. Ahmed, "Symmetric Key Generation Algorithm Using Sum of Subset N-P Problem," 2013.

[4] J. Rudinger and A. Finger, "Algorithm Design and Side Channel Vulnerability on the Example of DPA Attack," *Proceedings of the Sixth International Conference on Networking*, pp. 99, 2007.

[5] C. Y. Yan and R. H. Xiao, "Study of Block Algorithms Implement on Hardware in Information Security System," *IEEE International Conference on Business Management and Electronic Information*, Vol. 4, pp. 589-593, 2011.

[6] D. Anand, V. Khemchandani and R. K. Sharma, "Identity-Based Cryptography Techniques and Applications," *5th International Conference on Computational Intelligence and Communication Networks*, pp. 343-348, 2013.

[7] F. Fakhar and M. A. Shibli, "Management of Symmetric Cryptographic Keys in Cloud Based Environment," *International Conference on Advanced Communication Technology*, pp. 39-44, 2013.

[8] G. Murphy, A. Keeshan, R. Agarwal and E. Popovici, "Hardware-Software Implementation of Public-Key Cryptography for Wireless Sensor Networks," *ISSC, Dublin Institute of Technology*, pp. 463-468, 2006.

[9] H. Krawczyk, "The order of encryption and authentication for protecting communications (Or: how secure is SSL?)," Full version: http://eprint.iacr.org/2001.

[10] A. Tannous, J. Trostle, M. Hassan, S. E. McLaughlin and T. Jaeger, "New Side Channels Targeted at Passwords," *Annual Computer Security Applications Conference*, pp. 45-54, 2008.

[11] Y. Karandikar, X. Zou and Y. Dai, "An Effective Key Management Approach to Differential Access Control in Dynamic Environments," *Journal of Computer Science*, Vol. 2, No. 6, pp. 542-549, 2006.

[12] W. He, Y. Huang, K. Nahrstedt and W. C. Lee, "SMOCK: A Self-contained Public Key Management Scheme for Mission-critical Wireless Adhoc Networks," *Proceedings of the Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 201-210, 2007.

[13] R. Devi, "Importance of Cryptography in Network Security," *International Conference on Communication Systems and Network Technologies*, pp. 462-467, 2013.

[14] A. M. Abiachi, F. Ahmad and K. Ruhana, "A Competitive Study of Cryptography Techniques over Block Cipher," *International Conference on Modelling and Simulation*, pp. 415-491, 2011.

[15] J. Ren and L. Harnsciences, "Generalized Ring Signatures," *IEEE Transactions on Dependable and Secure Computing*, Vol. 5, No. 3, pp. 155-163, 2008.

[16] J. Lan, W. Goh, Z. Kong and K. Seng Yeo, "A Random Number Generator for Low Power Cryptographic Application," *International SoC Design Conference*, pp. 328-331, 2010.

[17] J. P. Delgrande, A. Hunter and T. Grote, "Representation and Verification of Cryptographic Protocols in a Theory of Action," *International Conference on Privacy, Security and Trust*, pp. 39-45, 2010.

[18] J. Lo, M. Hwang and C. Liu, "An efficient key assignment scheme for access control in a large leaf class hierarchy," *Journal of information sciences Elsevier science publisher*, Vol. 181, No. 4, pp. 917-925, 2010.

[19] K. Tiri, "Side-Channel Attack Pitfalls," *IEEE Design Automation Conference*, pp. 15-20, 2007.

[20] K. K. Khaing and K. M. Aung, "Secured Key Distribution Scheme for Cryptographic Key Management System," *International Conference on Availability, Reliability and Security*, pp. 481-486, 2010.

[21] K. Bhatele, A. Sinhal and M. Pathak, "A Novel Approach to the Design of a New Hybrid Security Protocol Architecture," *IEEE International Conference on Advanced Communication Control and Computing Technologies*, pp. 429-433, 2012.

[22] K. Sakiyama, Y. Li, K. Ohta and M. Iwamoto, "Information Theoretic Approach to Optimal Differential Fault Analysis," *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 1, pp. 109-120, 2012.

[23] K. N. Divya, D. S. Lakshmi and K. Vijaya, "A Routing-Driven elliptic Curve cryptography Based Key Management Scheme for Heterogeneous Sensor Networks," *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, No. 9, pp. 442-449, 2012.

[24] K. K. Pandey, V. Rangari and S. K. Sinha, "An Enhanced Symmetric Key Cryptography Algorithm to Improve Data Security," *International Journal of Computer Applications*, Vol. 74, No. 20, pp. 29-33, 2013.

[25] L. Harn and H. Lin, "A Cryptographic Key Generation Scheme for Multilevel Data Security," *Journal of Computers & Security, Elsevier Science Publishers*, Vol. 9, No. 6, pp. 539-546, 1990.

[26] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proceedings of the ACM conference on Computer and Communications Security*, pp. 41-47, 2002.

[27] L. J. Villalba, J. G. Matesanz, D. R. Canas and A. L. S. Orozco, "Secure Extension to the Optimized Link State Routing Protocol," *IET Information Security*, Vol. 5, No. 3, pp. 163-169, 2011.

[28] L. Uhsadel, M. Ullrich, A. Das, D. Karaklajic, J. Balasch, I. Verbauwhede and W. Dehaene, "Teaching HW/SW Co-Design With a Public Key Cryptography Application," *IEEE Transactions on education*, Vol. 56, No. 4, pp. 478-483, 2013.

[29] A. Biri, A. Ahmad, H. Afifi and D. Zeghlache, "Trade-off between Identity-Based and Certificateless Cryptography for future Internet," *IEEE International Symposium on Mobile Radio Communications*, pp. 2866-2870, 2009.

[30] M. I. Salam, P. Kumar and H. J. Lee, "An Efficient Key Pre-distribution Scheme for Wireless Sensor Network Using Public Key Cryptography," *International Conference on Networked Computing and Advanced Information Management*, pp. 402-407, 2010.

[31] M. Wang, C. Su, C. Horng, C. Wu and C. Huang, "Single and Multi-core Configurable AES Architectures for Flexible Security," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 18, No. 4, pp. 541-552, 2010.

[32] M. B. Shemaili, C. Y. Yeun, K. Mubarak and M. J. Zemerly, "A New Lightweight Hybrid Cryptographic Algorithm for The Internet of Things," *International Conference for Internet Technology and Secured Transactions*, pp. 87-92, 2012.

[33] N. Koblitz, "Elliptic Curve Cryptosystems," *Journal of Mathematics of Computation. American Mathematical Society*, Vol. 48, No. 177, pp. 203-209, 1987.

[34] N. B. Puhan and A. T. S. Ho, "Binary Document Image Watermarking for Secure Authentication Using Perceptual Modeling," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 393-398, 2005.

[35] N. Moebius, K. Stenzel and W. Reif, "Generating Formal Specifications for Security-Critical Applications," *Proceedings of the ICSE Workshop on Software Engineering for Secure Systems*, pp. 68-74, 2009.

[36] N. Lalithamani and Dr. K. P. Soman, "Towards Generating Irrevocable Key for Cryptography from Cancelable Fingerprints," *IEEE International Conference on Computer Science and Information Technology*, pp. 563-568, 2009.

[37] N. Khanna, J. Nath, J. James, A. Chakrabarti, S. Chakraborty and A. Nath, "New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm," *International Conference on Communication Systems and Network Technologies*, pp. 125-130, 2011.

[38] S. Ariffin, R. Mahmod, A. Jaafar and M. R. K. Ariffin, "Immune systems approaches for cryptographic algorithm," *International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 231-235, 2011.

[39] P. Gope, A. Singh, A. Sharma and N. Pahwa, "An Efficient Cryptographic Approach for Secure Policy Based Routing," *IEEE International Conference on Electronics Computer Technology*, pp. 359-363, 2011.

[40] S. S. Selvi, S. S. Vivek, S. Gopinath and C. Rangan, "Identity Based Self Delegated Signature - Self Proxy Signatures," *Fourth International Conference on Network and System Security*, pp. 568-573, 2010.

[41] R. M. Davis, "The Data Encryption Standard in Perspective," *IEEE Communications Society Magazine*, Vol. 16, No. 6, pp. 5-9, 1978.

[42] R. Dutta, Y. D. Wu and S. Mukhopadhyay, "Constant Storage Self-Healing Key Distribution with Revocation in Wireless Sensor Network," *IEEE International Conference on Communication*, pp. 1323-1328, 2007.

[43] R. S. Owor and J. Hamilton, "An Elliptical Cryptographic Algorithm for RF wireless devices," *Proceedings of the 2007 Winter Simulation Conference*, pp. 1423-1429, 2007.

[44] R. Pletka and C. Cachin, "Cryptographic Security for a High-Performance Distributed File System," *IEEE Conference on Mass Storage Systems and Technologies*, pp. 227-232, 2007.

[45] R. Azarderakhsh, A. R. Masoleh, and Z. Abid, "A Key Management Scheme for Cluster Based Wireless Sensor Networks," *International Conference on Embedded and Ubiquitous Computing*, pp. 222-227, 2008.

[46] R. S. Vignesh, S. Sudharssun and K. J. J. Kumar, "Limitations of Quantum & The Versatility of Classical Cryptography: A Comparative Study," *Second International Conference on Environmental and Computer Science*, pp. 333-337, 2009.

[47] Recommendation for key management part 1, July 2012, [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf.

[48] J. H. Li, R. Levy, M. Yu, and B. Bhattacharjee, "A Scalable Key Management and Clustering Scheme for Ad Hoc Networks," *Proceedings of the First International Conference on Scalable Information Systems*, pp. 1-10, 2006.

[49] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM Magazine*, Vol. 21, No. 2, pp. 120-126, 1978.

[50] X. Li, W. Zhang, X. Wang and M. Li, "Novel Convertible Authenticated Encryption Schemes without Using Hash Functions," *IEEE International Conference on Computer Science and Automation Engineering*, pp. 504-508, 2012.

[51] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu and T. La Porta, "Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks," *IEEE Transactions on Mobile Computing*, Vol. 6, No. 6, pp. 663-667, 2007.

[52] S. Nithyanandam, K. S. Gayathri, K. Raja and P. L. K. Priyadarshini, "Recent Trends In Secure Personal Authentication For Iris Recognition Using Novel Cryptographic Algorithmic Techniques," *IEEE International Conference on Process Automation, Control and Computing*, pp. 1-6, 2011.

[53] S. Verma, R. Choubey and R. Soni, "An Efficient Developed New Symmetric Key Cryptography Algorithm for Information Security," *International Journal of Emerging Technology and Advanced Engineering*," Vol.2, No. 7, pp. 18-21, 2012.

[54] S. Ju, "A Lightweight Key Establishment in Wireless Sensor Network Based on Elliptic Curve Cryptography," *IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, pp. 138-141, 2012.

[55] T. Teerakanok and S. Kamolphiwong, "Accelerating Asymmetric-Key Cryptography using Parallel-key Cryptographic Algorithm," *IEEE International Conference on Electronics, Computer, Telecommunications and Information Technology*, pp. 812-815, 2009.

[56] T. S. Halkidis, N. Tsantalis, A. Chatzigeorgiou and G. Stephanides, "Architectural Risk Analysis of Software Systems Based on Security Patterns," *IEEE Transactions on Dependable and Secure Computing*, Vol. 5, No. 3, pp. 129-142, 2008.

[57] V. Gupta, G. Singh and R. Gupta, "A Hyper Modern Cryptography Algorithm to Improved Data Security," *International Journal of Computer Science & Communication Networks*, Vol. 1, No. 3, pp. 258-263.

[58] J. J. Hoch and A. Shamir, "Fault Analysis of Stream Ciphers," *IEEE Transactions on Proceedings of Crypto'93, Springer-Verlag*, pp. 22-39, 1993.

[59] W. K. Koo, H. Lee, Y. Kim and D. H. Lee, "Implementation and Analysis of New Lightweight Cryptographic Algorithm Suitable for Wireless Sensor Networks," *International Conference on Information Security and Assurance*, pp. 73-76, 2008.

[60] X. He, M. Zhang and Q. Yang, "A Storage Performance Evaluation Kernel Module for Block-Level Storage Systems under Faulty Conditions," *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, No. 2, pp. 38-149, 2005.

[61] X. Du, M. Guizani. Y, Xiao. and H. Chen., "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 8, No. 3, pp. 1223-1229, 2009.

[62] X. Dongyang, Z. Tang and Y. Yinyan, "An Efficient Key Management Scheme for Segment based Document Protection," *IEEE Conference on Consumer Communications and Networking Conference*, pp. 896-900, 2011.

[63] S. Jailin, R. Kayalvizhi and V. Vaidehi, "Performance Analysis of Hybrid Cryptography for Secured Data Aggregation in Wireless Sensor Networks," *IEEE-*

International Conference on Recent Trends in Information Technology, pp. 307-312, 2011.

[64] X. Li, L. Yu and L. Wei, "The Application of Hybrid Encryption Algorithm in Software Security," *IEEE International Conference on Consumer Electronics, Communications and Networks*, pp. 669-672, 2013.

[65] H. Wang and S. Han, "A Provably Secure Threshold Ring Signature Scheme in Certificateless Cryptography," *International Conference of Information Science and Management Engineering*, pp. 105-108, 2010.

[66] Y. Bharadwaj and S. Chakraverty, "A Design Pattern for Symmetric Encryption," *International Conference on Control, Computing, Communication and Materials*, pp. 1-6, 2013.

[67] Z. Huawei, Q. Jing and F. Zhifeng, "Converters for Designing Applied Cryptographic Protocols", *International Conference on Embedded and Ubiquitous Computing*, pp. 139-143, 2008.

[68] Zhihua NIU, "Research on Excellent Periodic Binary Sequence with Genetic Algorithm," *IEEE International Conference on Computer and Information Technology*, pp. 355-358, 2012.