A

Thesis Report

On

Optimized Encryption Algorithm using Dynamic Keys

Submitted towards the fulfillment of requirement for the award of degree of

MASTER OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Hitesh Mittal

Roll. No. 801261010

Under the guidance of

Dr. Ajay Kakkar

(Assistant Professor)

Electronics and Communication Engineering Department

THAPAR UNIVERSITY

PATIALA-147004

Declaration

I hereby declare that the work which is being presented in this thesis entitled "**Optimized Encryption Algorithm using Dynamic Keys**" in partial fulfillment of the requirement for the award of degree of ME (Electronics and Communication Engineering) at Thapar University, Patiala is an authentic record of my study carried out under the supervision of Dr. Ajay Kakkar (Assistant Professor), ECED during the year 2013-2014.

Date: 14/07/2014

Hitesh Mittal
Roll No.801261010

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

Date: 14/7/14

Dr. Ajay Kakkar
Assistant Professor
ECED, Thapar University

Countersigned By:

Dr. Sanjay Sharma
Head of Department
ECED, Thapar University

Dr. S. K. Mohapatra
Dean of Academic Affairs
Thapar University

Acknowledgement

First of all, I would like to express my gratitude to **Dr. Ajay Kakkar**, **Assistant Professor,** Electronics & Communication Engineering Department, Thapar University, Patiala for his patient guidance and support throughout the thesis. I am truly very fortunate to have the opportunity to work with him. I found this guidance to be extremely valuable. I am also thankful to **Dr. Sanjay Sharma, Professor and Head of Department** as well as PG coordinator **Dr. Kulbir Singh, Associate Professor**, Electronics and Communication Engineering Department. I would like to thank entire faculty and staff of Electronics and Communication Engineering Department and then friends who devoted their valuable time and help me in all possible ways towards successful completion of this work. I thank all those who have contributed directly or indirectly to this work.

Lastly, I would like to thanks my parents for their years of unyielding love and encourage they have always wanted the best for me and I admire their determination and sacrifice.

Hitesh Mittal

Abstract

Data security is an essential component of an organization in order to keep the information safe from various competitors. It helps to ensure the privacy of a user's personal information from others. Secured and timely transmission of data is always an important aspect for an organization. Strong encryption algorithms and optimized key management techniques always help in achieving confidentiality, authentication and integrity of data and reduce the overheads of the system. Keeping in view the importance of dynamic keys for secure data transmission, the work is focused on the use of dynamic keys for data security. In this work various encryption algorithms have been studied. Literature Survey has been carried out by incorporating key papers related to data encryption. From the literature survey; gaps and observation have also been drawn. RSA is critically analyzed and key management has also been optimized. Simulation results have been achieved using MATLAB 7.3; results prove that the proposed algorithm is optimized compared to RSA in terms of hacking and processing time. Finally, conclusion and future work has also been stated at the end of thesis.

Table of Contents

List of Abbreviations

CT	Cipher Text
PT	Plain Text
TDES	Triple Data Encryption Standard
PGP	Pretty Good Privacy
DES	Data Encryption Standard
AES	Advanced Encryption Standard
RSA	Rivest Shamir Adleman
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
DSA	Digital Signature Algorithm
HA	Hash Algorithm
MA	Modular Arithmetic
PKCS	Public Key Cryptographic Standards
TEK	Traffic Encryption Key
ECB	Electronic Codebook
CC	Central Controller
CBC	Cipher Block Chaining
REK	Resource Encryption Key
RL	Resource Lists
CA	Certificate Authority
DIFAC	Differential Access Control

List of Figures

List of Tables

List of Graphs

List of Publications

[1] H. Mittal and A. Kakkar, "Performance Analysis of Multiple Keys used for Data Security", International Journal of Computer Applications, Vol. 95, No. 14, pp. 29-32, 2014.

Chapter 1: Introduction to Cryptography

1.1 Overview:

Cryptography is a technique used to avoid unauthorized access of data. It has two main components; a) Encryption algorithm, and b) Key. Sometime, multiple keys can also be used for encryption. A number of cryptographic algorithms are available in market such as DES, AES, TDES and RSA. The strength of these encryption algorithms depends upon their key strength. Strong encryption algorithms and optimized key management techniques always help in achieving confidentiality, authentication and integrity of data and reduce the overheads of the system. The long key length takes more computing time to crack the code and it becomes difficult for the hacker to detect the cryptographic model. Cryptography is basically divided into two categories; a) Symmetric Cryptography, and b) Asymmetric Cryptography. In symmetric cryptography the key used to encrypt the message is the same as the key decrypting the message whereas in asymmetric cryptography different key is used for encryption and decryption. Asymmetric algorithms are relatively slower than symmetric algorithms but provide a good security level. In cryptography there are some important terms and are given below:

- Plaintext: It is the original text which has to be encrypted [3].
- Cipher text: It is the encrypted text. The text obtain after encoding the data with the help of a key is known as cipher text [40].
- Key: It is a word or value that is used to encrypt the plain text or decrypt the cipher text [37].
- Encryption: The method of converting the data into coded form with the help of key is called encryption [4].
- Decryption: The method of converting the encoded data to the original form is called decryption [37].
- Crypto Analyst: A crypto analyst is a person who is an expert in analyzing and breaking codes [3].

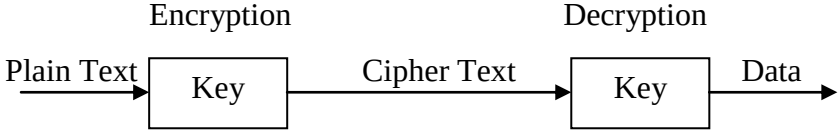


Figure 1.1: Cryptographic Model [3]

1.2 Need of Cryptography:

In today's world cryptography has become a necessity for all the organizations. Data security is an essential component of an organization in order to keep the information safe from various competitors. It also helps to ensure the privacy of a user from others. These days passwords are not considered as reliable for this task because it is easy to guess passwords due to its short range. Moreover, if the range of password is small a brute force search can be applied to crack it [3]. So, as to protect our data various algorithms have been designed. It helps us to securely access bank accounts, electronic transfer of funds and many more daily life applications.

1.3 Type of Cryptography:

There are basically two types of Cryptography; a) Symmetric Cryptography, and b) Asymmetric Cryptography.

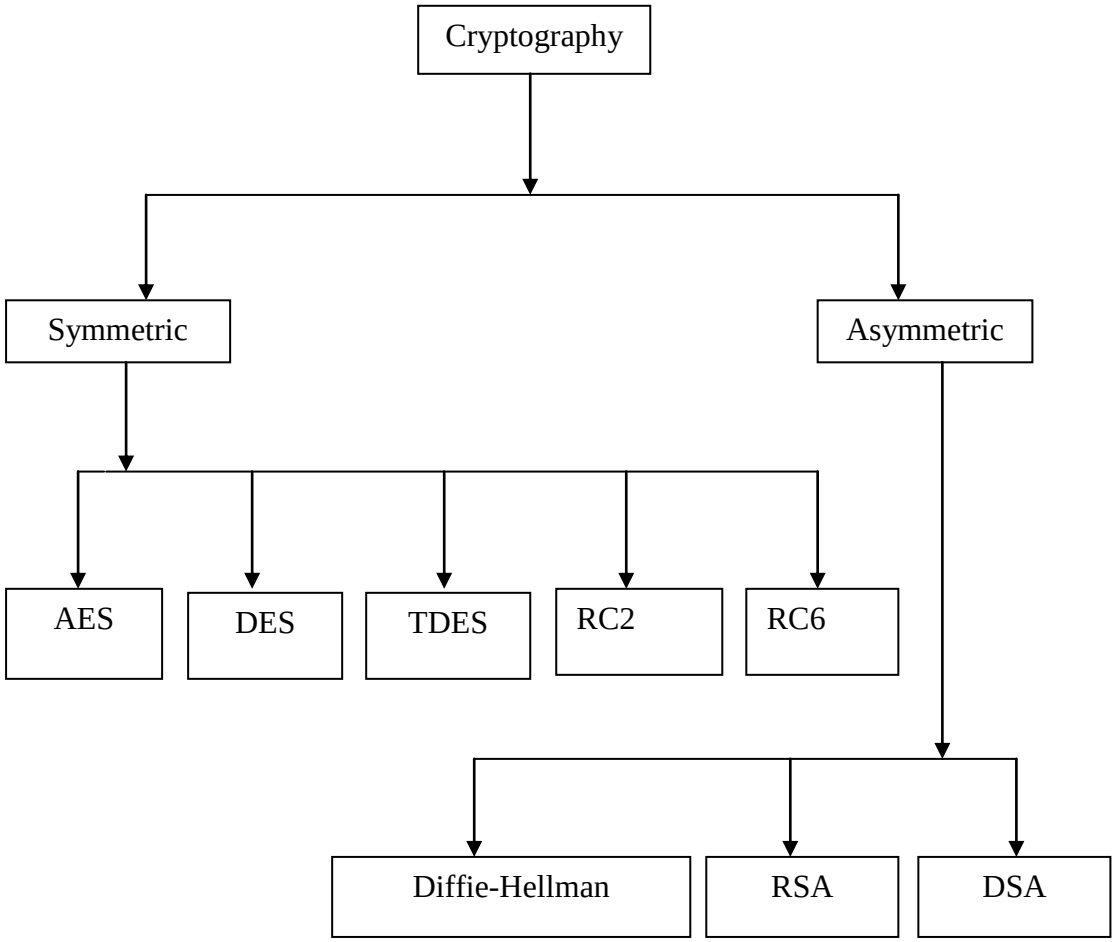


Figure 1.2: Classification of Cryptography [37]

1.3.1 Symmetric (Private) Cryptography:

The private cryptography is an encryption process where key used to encrypt the message is the same as the key decrypting the message. Private key cryptography is fast and efficient, making it ideal for large data transmissions [3]. Private key cryptography is more effective when used with public key cryptography because it is much faster. Few common types of symmetric encryption are briefly explained below:

a) Data Encryption Standard (DES):

DES was developed by IBM but was later adopted by the US Government as a National Standard [41]. It divides the original message into 64-bit blocks. Each block is then permutated to change the order of its bits. It divides the 56-bit key into two 28-bit halves. Each half is than circular-shirted to the left, reconnected and enlarged to 48 bits. Then the right half of the plaintext blocks is expanded to 48-bits. Then the new 48-bit plaintext block is crossed over with the 48-bit key. This result is then converted to 32-bits using a substitution function. This 32-bit block is crossed over with the left half of the plaintext block forming two new 32-bit halves.

b) Triple Data Encryption Standard (TDES):

Triple DES goes through 3 iteration of DES effectively encrypting data with a 168-bit key which is strong enough to secure sensitive information [12]. The data is first encrypted using a 56-bit DES key, decrypted with another 56-bit DES key, and finally encrypted again with the original 56-bit DES key. 3 DES contains several levels of encryption and it can better protect against middle attacks.

c) Advanced Encryption Standard (AES):

The AES algorithm accepts a block size of 128 bits [4]. Depending on key size the name of the standard is modified to AES-128, AES-192 or AES- 256 respectively. A number of AES parameters depend on the key length. For example, if the key size used is 128, the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively.

d) RC2:

In cryptography RC2 is a symmetric key block cipher designed in 1987 [40]. RC2 is a 64-bit block cipher with a variable size key. Its 18 rounds are arranged as a source-heavy

unbalanced Feistel network, with 16 rounds of one type of mixing punctuated by two rounds of another type mashing.

e) RC6:

RC6 is a new block cipher submitted to NIST for consideration as the new AES [9]. It has three components; a key expansion algorithm, an encryption algorithm and a decryption algorithm. It makes use of data-dependent rotations, similar to DES rounds. RC6 has provided a simple cipher which yields to numerous evaluations and adequate security in a small package.

1.3.2 Asymmetric (Public) Cryptography:

The public cryptography is an encryption process where key used to encrypt the message is different from key decrypting the message. There exist two keys, a) private key, and b) public key. Anyone may have access to the public key but the private key is kept secret. When one is used for encryption, the other is used for decryption [3]. In this scheme encryption performed with the public key ensures secrecy of the message, since no one other than the person having the private key can decrypt it. Few common types of asymmetric encryption are briefly explained below:

a) Diffie - Hellman Algorithm:

Diffie–Hellman key exchange is a specific method of exchanging cryptographic keys proposed in 1976 [40]. It allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key is used to encrypt subsequent communications using a symmetric key cipher.

b) RSA:

RSA algorithm is designed by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT [3]. It has two keys; public key and private key. Both keys are used for encryption and decryption purpose. Sender encrypts the message using receiver public key and when the message gets transmit to receiver, then receiver can decrypt it using his own private key. It uses two prime numbers p and q to generate private key and public key. The security of RSA depends on the product of these two numbers represented by n.

c) Digital Signature Algorithm (DSA):

A digital signature algorithm is a public key cryptographic algorithm designed for authenticating digital message [40]. A message is signed by a secret key to produce a signature, and then this is verified against the message by a public key. Any party can verify the signatures but only one party with the secret key can sign the messages. A valid digital signature gives recipient a reason to believe that the message was created by a known sender. He possesses the secret key, and that it was not altered in transit.

1.4 Types of Attacks on Cryptography:

There are basically two types of attack. One is on system and other is on data shown in figure.

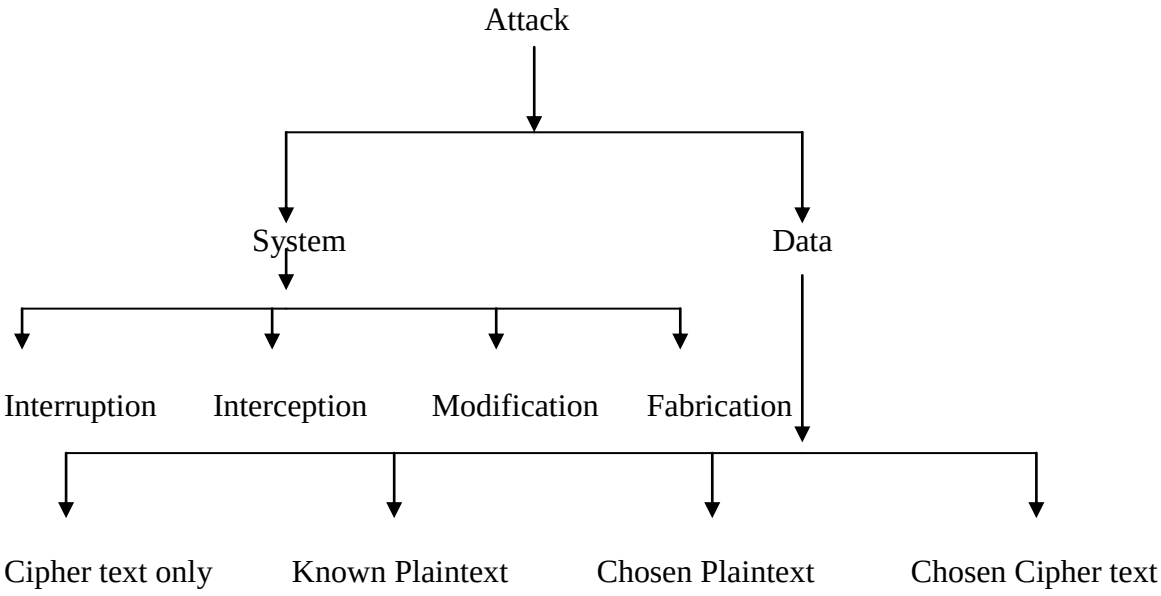


Figure 1.3: Classification of Attacks on Cryptography

1.4.1 System Attacks:

In general there is a flow of information from a source to a destination. The attacks which are on the flow of information are known as system attacks. The main security threats are listed below:

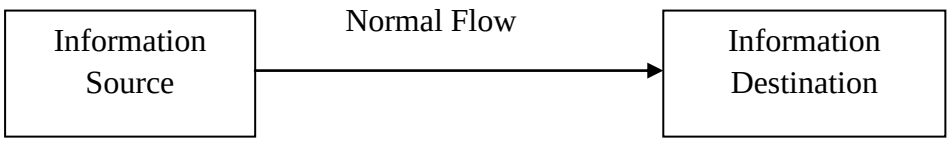


Figure 1.4: Normal Flow of Information

- **Interruption:** It is an attack on availability of the resource. When the data flowing through source to destination becomes unavailable or unusable [16].

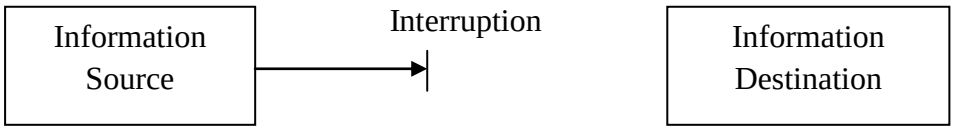


Figure 1.5: Interrupted Data Flow

- **Interception:** It is an attack on the confidentiality of the system. In this attack an unauthorized party also has the access to a model. A person, program and a computer may be the unauthorized party [37].

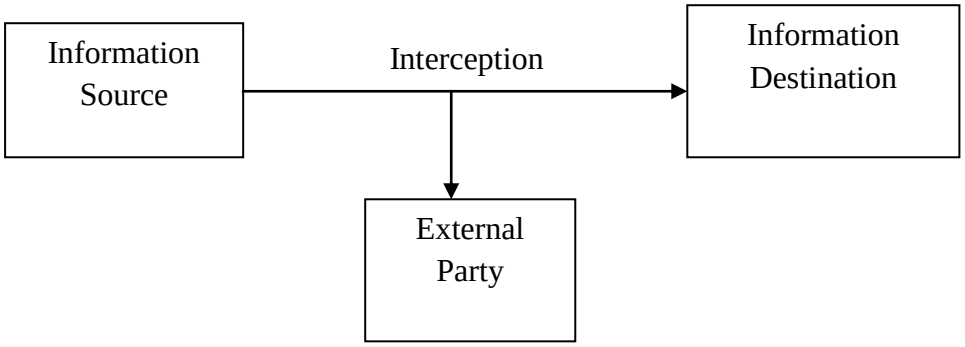


Figure 1.6: Interception Attack

- **Modification:** It is an attack on integrity of the system. In this attack an unauthorized party not only has the access to an asset but has the power to modify it [41].

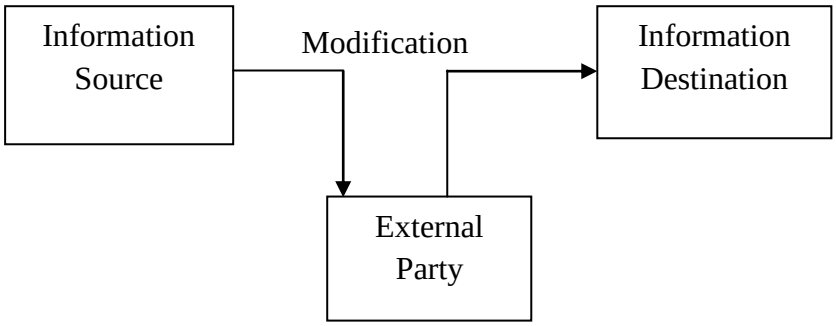


Figure 1.7: Modification of Data

- **Fabrication:** It is an attack on authenticity of the system. In it an unauthorized party inserts counterfeit objects into the system [37].

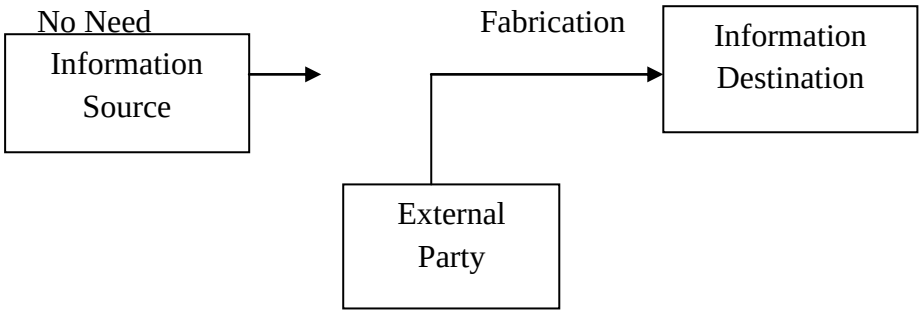


Figure 1.8: Fabrication System Attack

1.4.2 Data Attacks:

An attempted crypto analysis is known as an attack. The level of information that decoder is able to extract from the cryptosystem and can be divided into five ways of decryption which are as follows:

- **Cipher text only attack:** The crypto analyst has cipher text of several messages and all of which were encrypted using the same encryption algorithm. Then job is to recover the plain text or the key used to encrypt the messages. So, to decrypt other part of messages encrypted with the help of same keys [37].
- **Known Plaintext attack:** Crypto analysts seek the possession of pairs of known plain text and cipher text. Then job is to hold the key used to encrypt the messages or an algorithm to decrypt messages [3].
- **Chosen Plaintext Attack (CPA):** Crypto analyst not only hold the cipher text but also some parts of chosen plain text. Intruder is identified to be placed at encryption site to do the attack [37].
- **Chosen cipher text attack (CCA):** In this crypto analyst hold the possession of chosen cipher text and plain text being decrypted from the private key. However, it only has access to an encryption machine [11].

1.5 Outcome:

This chapter includes various definition, parameters and types of cryptography. Brief introduction about various attacks generated by hackers has also been stated.

1.6 Organization of Thesis:

Chapter 1 shows the overview and types of cryptography. A brief introduction about various attacks generated by hackers has been included in this chapter. Chapter 2 includes literature survey. It involves the work done by the various researchers in the field of cryptographic algorithm for data security. From the literature survey various observations have been drawn and listed at the end of this chapter. From the observation various objectives have also been derived. Chapter 3 includes detailed description and various security issues related to RSA. After studying security issues related to RSA a new algorithm is proposed which is stated in this chapter. Chapter 4 includes comparison between proposed approach and other algorithms. Chapter 5 summarizes the simulation result using MATLAB 7.3. Finally, Chapter 6 suggests conclusion and future scope of the work done in this thesis.

CHAPTER 2: Literature Survey

This chapter involves the work done by the various researchers in the field of cryptographic algorithm for data security. From the literature survey various observations have been drawn and listed at the end of this chapter. From the observations various objectives have also been drawn.

2.1 Literature Survey:

Martin E. Hellman [22] extended the shannon theory approach to cryptography. He discussed about Shannon's random cipher model was conservative than in such case when a randomly chosen cipher was considered, the security of model falls significantly. The concepts of matching a cipher to a language and the trade-off between local and global uncertainty were also developed. The limitation of this approach is that it is not directly applicable to designing practical cryptographic systems.

H. C. Williams [10] modified the RSA public-key encryption algorithm. He suggested that if the encryption procedure was broken into a certain number of operations than remainder used as modulus could be factored after few more operations. This technique was in similar appearance to RSA. The main limitation of this scheme was that very large prime numbers were used and generated mathematical errors were observed.

Taher Elgamal [34] proposed a signature scheme based on discrete logarithms and implemented Diffie-Hellman key distribution scheme that achieves a public key cryptosystem. The security of both systems relies on the difficulty of computing discrete logarithms over finite fields.

Adam J. Elbirt *et al.* [1] evaluated the AES block cipher algorithm using FPGA based kit. They proposed that reprogrammable devices such as field-programmable gate arrays (FPGAs) are highly attractive options for hardware implementations of encryption algorithms. Proposed cryptographic algorithm had physical security and potential which has much higher performance than software solutions. The main limitation was that when the size of implementation increases then the number of rounds unrolled or pipelined was

increased and this increase was partially offset by the packing of the round keys within the round structure.

M.A. Hasan *et al.* [23] discussed the concept of power analysis attacks and algorithmic approaches used for counter measures for koblitz curve cryptosystems. Power analysis attacks were applied to cryptosystems that uses scalar multiplication on koblitz curves. Simple and the differential power analysis attacks both were considered and a number of countermeasures were suggested. While the proposed countermeasures against the simple power analysis attacks rely on making low power consumption for the elliptic curve scalar multiplication which was independent of the secret key. But for the differential power analysis attacks depend on randomizing the secret key prior to each execution of the scalar multiplication.

SuKyoung Lee *et al.* [29] proposed a hierarchical restoration scheme for handling multiple failures in GMPLS networks. This scheme was used where hierarchical Shared Risk Link Groups were applied. They introduced backup group multiplexing into their hierarchical scheme to precipitate the restoration of multiple label switched paths with failures. Proposed scheme selects a backup path with enough resources to satisfy renegotiated Quality of Service.

Haowen Chan *et al.* [9] worked on random key pre-distribution scheme used for Wireless Sensor Networks. They presented three mechanisms for key establishment using the framework of pre-distributing a random set of keys to each node. a) In the q-composite keys scheme there was a trade off the unlikeliness of a large-scale network attack in order to significantly strengthen random key redistributions strength against smaller-scale attacks. b) In the multipath-reinforcement scheme, they showed that how to strengthen the security between any two nodes by leveraging the security of other links. c) They presented the random-pair wise keys scheme, which perfectly preserves the secrecy of the rest of the network when any node was captured. It enables the node-to-node authentication and quorum-based revocation. The major limitation of these schemes was that they provide security to a particular parameter at the cost of other.

Chih-pin Su *et al.* [6] designed an AES processor which has high-throughput and low-cost. They proposed an efficient hardware implementation of the AES with key expansion capability. This transformation technique reduced the hardware overhead of the S-Box by 64%.

Stefan Mangard *et al.* [31] proposed a highly regular and scalable AES hardware architecture which was suited for full-custom as well as for semicustom design flows. This architecture was scalable in terms of throughput and in terms of the used key size. Similarities of encryption and decryption were utilized to provide a high level of performance by using only a relatively small area. This performance was reached by balancing the combinational paths of the design.

Lemma Hundessa *et al.* [20] provided a mechanism to obtain optimal and alternative LSP for multiple failures. The mechanism was able to handle multiple failure in LSP for critical traffic than the alternative LSP established. The routing decision was taken close to the point of failure which reduces the restoration time. The use of pre-established alternative LSP also reduces the restoration time and avoids blocking when looking for an alternative path. The disadvantage of using a preplanned alternative LSP was that it may not be the optimal one at the time of failure.

Bharat B. Madan *et al*. [4] worked on various methods used for modeling and quantifying the security attributes of intrusion tolerant systems**.** Various issues related to quantifying the security attributes of an intrusion tolerant system were also addressed. Response of a security intrusion tolerant system to an attack was modeled as a random process. They facilitates the use of stochastic modeling techniques to predict the attacker behavior. They had also computed a security measure called the mean time to security failure and also compute probabilities of security failure due to violations of different security attributes.

Hung-Yu Chien [12] presented an efficient time bound hierarchical key assignment scheme. They proposed a tamper-resistant device which has a new time-bound key assignment scheme. It significantly improves the computational performance and reduces the implementation cost.

Ho Won Kim *et al.* [13] designed and implemented a private and public key crypto processor and its application to a Security System. A special-purpose microprocessor was optimized for the execution of cryptography algorithms. This crypto processor could be used for various security applications such as storage devices, embedded systems, network routers, security gateways using IP Sec and SSL protocol, etc. They had presented the design and implementation of a crypto processor composed to a 32-bit RISC processor and coprocessor blocks dedicated to the AES, KASUMI, SEED, triple-DES, ECC and RSA crypto algorithms.

Xinmiao Zhang *et al.* [38] worked on High-Speed VLSI Architectures for the AES Algorithm. They presented novel high-speed architectures for the hardware implementation of the Advanced Encryption Standard algorithm. The proposed design employs combinational logic. As a direct consequence the unbreakable delay incurred by look-up tables in the conventional approaches was eliminated and the advantage of sub pipelining could be further explored. The composite field arithmetic was employed to reduce the area requirements.

Yanchao Zhang *et al.* [40] worked on Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks. They worked on the notion of location-based keys by binding private keys of individual nodes to both their IDs and geographic locations. They developed LBK-based neighborhood authentication scheme to localize the impact of compromised nodes to their vicinity. The conclusion was that they presented a comprehensive set of location-based compromise tolerant security mechanisms for WSNs.

Patrick Traynor *et al.* [27] worked on Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. They demonstrated a probabilistic unbalanced distribution of keys throughout the network that leverages the existence of a small percentage of more capable sensor nodes can not only provide an equal level of security, but also reduce the consequences of node compromise. To fully characterize the effects of the unbalanced key management system they designed, implemented and measured the performance of a complementary suite of key establishment protocols. The pre deployed

keys were used for nodes operating in isolation from external networks can securely and efficiently establish keys with each other.

Hung-Min Sun *et al.* [15] proposed dual RSA algorithm and also analyzed the security of the algorithm. They presented new variants of RSA whose key generation algorithms output two distinct RSA key pairs having the same public and private exponents Two applications for Dual RSA were blind signatures and authentication. The security of Dual RSA was raised in comparison to RSA when there were small values of e and d. The main disadvantage of using dual RSA was that the computational complexity of the key generation algorithms was also increased.

Hua Li *et al*. [11] worked on new compact dual-core architecture used in AES. They presented a new compact architecture which consists of two independent cores that encryption and decryption simultaneously. Proposed key generation unit with 32-bit data path was also explored in order to provide round keys for encryption and decryption process. A novel way to implement shift rows shift rows one of the key designs in the compact 32-bit architecture was also proposed in order to increase encryption time. The major limitation was that this design also requires fewer more hardware resources in comparison to the designs.

Jason H. Li *et al.* [16] worked on scalable key management and clustering scheme for secure group communications in *Adhoc* and WSN. They describe scalable key management and clustering to achieve more secured system. The scalability problem was solved by partitioning communicating devices into subgroups with a leader in each subgroup. The Distributed Efficient Clustering Approach (DECA) provided robust clustering to form subgroups and simulation results demonstrate that DECA is energy efficient and resilient against node mobility. This scheme was not suitable for large cluster size.

Jong Tae Park [17] presented a dynamic path management strategy using resilience constraints under multiple link failures in MPLS/GMPLS networks. The path recovery mechanism could rapidly find an optimal backup path which satisfied the resilience constraints under multiple link failure occurrences. He developed a decomposition

theorem and backup path construction algorithm for the fast restoration of resilience-guaranteed backup primary path with an arbitrary configuration. We can extend this work by the combination of resilience constraints with other quality of service constraints such as bandwidth for optimal path management in MPLS/GMPLS networks.

Jian Ren [18] proposed a generalized ring signature scheme based on the original ElGamal signature scheme which was used to achieve unconditional signer ambiguity. The proposed scheme provides more security against adaptive chosen-message attack in the random oracle model. To overcome the limitation of this scheme a new scheme known as generalized multi-signer ring signature scheme was introduced to increase the level of confidence. The main limitation of this scheme was that the verifier was unable to indicate which member actually produced the signature.

Francesco Menichelli *et al.* [8] worked on high-level side-channel attack modeling and simulation which was used for security of critical systems on chips. They proposed an exploration approach centered at high-level simulation which was used to evaluate the actual implementation of a cryptographic algorithm. The simulation was performed within unified tool based on System C. It could be modeled as a software implementation running on a microprocessor-based architecture or a dedicated hardware implementation and mixed software-hardware implementation. In future work can be done to refine the model with information coming from post circuit and post layout design phases.

Elisa Bertino [7] worked on the concepts, approaches, and challenges of database security. Relevant concepts underlying the notion of database security and summarizes the most well-known techniques were also discussed which was focused on access control systems. He describes the key access control models which were mandatory access control models, and the role-based access control model. He also discussed security for advanced data management systems. The major limitation was that when an individual user wants to change the subscription a new device needs to be issued.

Spyros T. Halkidis *et al.* [30] analyzed the architectural risk software systems based on security patterns. The first step was to determine to what extent specific security patterns shield from known attacks. This information is fed to a mathematical model based on the

fuzzy-set theory and fuzzy fault trees in order to compute the risk for each category of attacks. The whole process was automated using a methodology that extracts the risk of a software system by reading the class diagram of the system under study. In this way security problems can be detected at an early stage that reduced the cost compared to the introduction of security during implementation. Extension to this work would be the automatic introduction of missing security patterns either at the design phase of a system being developed or in already implemented software systems.

Toshinori Fukunaga *et al.* [36] worked on practical fault attack on a cryptographic LSI with ISO/IEC 18033-3 block ciphers. They presented practical fault attack which results in six kinds of block ciphers listed in ISO/IEC 18033-3 that were implemented on an LSI: AES, DES, Camellia, CAST-128, SEED, and MISTY1. They developed an experimental environment that injects faults into any desired round by supplying a clock signal with a glitch. They examined practical attack assumptions and the fault model based on experimental results.

Aqeel Khalique *et al.* [2] worked on a password authenticated key agreement scheme based on ECC Using Smart Cards. It is one of the best public key techniques for its small key size and high security. It is also suitable for secure access of smart cards due to the implementation on smart cards.

Mao-Yin Wang *et al.* [24] configured single and multi-core AES architectures for flexible security. According to them the major building blocks for the architecture of AES were a group of AES processors. Each AES processor provides block cipher schemes with a novel key expansion design approach for the original AES algorithm. In this multi core architecture, the memory controller of each AES processor was designed for the maximum overlapping between data transfer and encryption thus reducing interrupt handling load of the host processor.

Mao-Yin Wang *et al.* [25] worked on A Mesh-Structured Scalable IPsec Processor that executed the IPsec protocols for Internet security gateway applications. They have developed several area-efficient cryptographic IPs embedded in MIPsec to lower silicon cost. Both handshake and contention issues were solved in the scheme, such that

performance can be scaled up. Specifically, the 6.23-million- gate MIPsec achieves 10-Gb/s wire speed for each routing direction. The proposed MIPsec is suitable for transport mode or other crypto mix as well.

Massimo Alioto *et al.* [26] worked on Differential Power Analysis Attacks to Precharged Buses: A General Analysis for Symmetric-Key Cryptographic Algorithms. They proposed a general model of multibit differential power analysis attacks to precharged buses. The main parameters that are of interest in practical DPA attacks were analytically derived under appropriate approximations and a novel figure of merit to measure the DPA effectiveness of multibit attack was proposed.

L.J. Garcia Villalba *et al*. [21] securely extended optimized link state routing protocol. Their study presented an extension of OLSR called COD-OLSR that provides security for OLSR in case of incorrect message generation attacks which can occur in two forms. This was one of its main features and was taken into account for current topology of node sending the message. The behavior of COD-OLSR against different attackers in a variety of situations is evaluated.

Hao Yang *et al.* [14] studied a case on DNSSEC so as to deploy cryptography in Internet-Scale Systems. They provide the first systematic examination of the design, deployment, and operational challenges encountered by DNSSEC over the years. Their study reveals a fundamental gap between cryptographic designs and operational Internet systems. They believe that the insights gained from this study can offer valuable inputs to future cryptographic designs for other Internet-scale systems. The limitation is that are not able to deploy cryptography in internet scale systems.

Septimiu Fabian Mare *et al.* [32] communicated data safely using Steganography, AES and RSA. They introduced a new secret data communication system that employs the usage of two state-of-the art cryptographic algorithms together with steganography. The joining of these three techniques builds a robust steganography-based communication system capable of withstanding multiple types of attacks, detection and reverse engineering. Their system was designed in a way that offers a solution to the major flaws presented in other steganography communication systems. Their new communication

model presents increased reliability because it covers key aspects in terms of data security are unidirectional encryption system and authenticity verification for both cover image and secret data and chained cryptographic systems and unidentifiable communication stream.

Shengrong Bu *et al.* [33] worked on Distributed Combined Authentication and Intrusion Detection with Data Fusion in High-Security Mobile Ad Hoc Networks. Multimodal biometrics was deployed to work with intrusion detection systems to alleviate the short comings of unimodal biometric systems. Each device in the network had measurement and estimation limitations, Observations of each device were fused and more than one device could be chosen using Dempster-Shafer theory for data fusion. The system decides whether user authentication was required and which biosensors should be chosen, depending on the security posture. The decisions are made in a fully distributed manner by each authentication device and IDS. Combining continuous authentication and intrusion detection could be an effective approach to improve the security performance in high-security MANETs.

Chong Hee Kim [5] improved differential fault analysis on AES key schedule. Proposed advanced encryption standard for which the main target is known DFA. Implementation of AES is known to be vulnerable to DFA which could be split into two categories depending on the fault location that has the DFA on the state and the DFA on the key schedule. The major limitation is that if the key schedule is not redone for recomputation then it cannot prevent DFA on the AES Key Schedule. The major problem was that if the key schedule was not done again for recomputation then it cannot prevent DFA on the AES Key Schedule.

Kirtiraj Bhatele *et al.* [19] designed of new hybrid security protocol architecture. They suggested that new security protocol for on-line transaction could be designed using combination of both symmetric and asymmetric cryptographic techniques known as Hybrid cryptography. This protocol serves three important cryptographic primitives; integrity, confidentiality and authentication. They encapsulated all the developments in the designing of new security protocol for On-line transaction and their importance was very much evident from the fact that communication has a major impact on today's

business. The proposed hybrid security protocol was more immune against the square attacks because of the inclusion of AES.

Yang Li *et al.* [39] worked on New Fault-Based Side-Channel Attack called fault sensitivity analysis attack Using Fault Sensitivity. They explained the successful FSA attacks against three Advanced Encryption Standard hardware implementations, where two of them were resistant to the differential fault analysis. They also discussed the countermeasures against the proposed FSA attacks. They proposed a new fault-based attack called FSA attack. The FSA attack was the first one that introduced the concept of fault injection intensity to the fault-based attacks. In the FSA attack, fault injections were used to test out the sensitive information leakage called fault sensitivity.

Zhiguo Wan *et al.* [41] worked on hierarchical attribute based solution for flexible and scalable access control in cloud computing. They proposed hierarchical attribute-set-based encryption by extending cipher text policy attribute-set-based encryption with a hierarchical structure of users. The proposed scheme not only achieves scalability due to its hierarchical structure but also inherits flexibility. It employed multiple value assignments for access expiration time to deal with user revocation more efficiently than existing schemes.

Tomasz Rams *et al.* [35] surveyed a group key distribution scheme with self-healing property. They analyzed and compare the most significant key distribution schemes by looking at the selective key distribution algorithms, at the redistributed secret data management, and the self-healing mechanisms. They reviewed polynomial-based algorithms, exponential arithmetic based algorithms, hash-based techniques, and others. Attention is paid to the self-healing property which permits group members to recover missing session keys from the recent key distribution broadcast messages without any additional interaction with the group manager. Limitation of the self-healing techniques adds some redundant information to the broadcast message so as to allow user nodes to recover previous session keys which were lost due to communication errors.

2.2 Gaps in study and observations:

From the previous section following observations have been drawn:

1) Single key of short length is not capable to provide secured cryptographic model.
2) Long length key can be able to provide secured cryptographic model.
3) In order to keep all the primitives in limit optimized hardware is required.
4) Use of dynamic keys are preferred for encryption process
5) There is a need to optimize the key arrangement in order to achieve secured cryptographic model.

2.3 Objectives:

From the observations it is concluded that there is a need to develop a cryptographic model based upon the dynamic keys for data security. So, objectives have been derived from the observations obtained from literature survey. The objectives are given as:

1) To study various encryption algorithm used for data security.
2) To study various attacks generated by hackers.
3) To study different soft computing tools used to achieve secured cryptographic model.
4) To design an optimized algorithm based upon dynamic keys for encryption.
5) Comparison of our approach with existing ones.

CHAPTER 3: RSA and Proposed Algorithm

This chapter includes detailed description and various security issues related to RSA. After studying security issues related to RSA a new algorithm is proposed which is stated in this chapter.

3.1 RSA Algorithm:

RSA algorithm is designed by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT in 1978 [3]. It can be broadly classified into three steps; a) key generation, b) encryption, and c) decryption. RSA has two keys; public key and private key. Both keys are used for encryption and decryption purpose. Sender encrypts the message using receiver's public key and when the message gets transmit to receiver, then receiver can decrypt it using his own private key. It uses two prime numbers p and q to generate private key and public key. For small values of p & q, the designing of key in the encryption process becomes too weak and one could be able to decrypt the data using random probability theory. On the other hand, if large value of p and q are selected, it consumes more time and its performance degrades. Hence, RSA is slower than other symmetric algorithms.

3.2 Practical Implementation of RSA Algorithm:

To implement RSA, one has to focus on three parts which are a) key generation, b) encryption process, and c) decryption process.

3.2.1 Key Generation Algorithm:

There are two types of keys in RSA; public key and private key. The steps for key generation are given as:

1) Generate two large prime numbers p and q.
2) Compute $n = p * q$
3) Compute $z = (p-1) * (q-1)$
4) Choose a number relatively prime to z and call it d.
5) Find e such that $e * d = 1 \, mod \, z$.
6) Public key is (n, e).
7) Private key is (n, d).

3.2.2 Encryption Algorithm:

In RSA encryption is done with the help of public key to generate the cipher text. The steps for encryption are given as:

1) Obtains the recipient public key (n, e).

2) Represents the plain text message as positive integer.

3) Compute the cipher text $c = m\hat{}e \; mod \; n$.

4) Sends the cipher text.

3.2.3 Decryption Algorithm:

In RSA decryption is done with the help of private key to get the plain text. The steps for decryption are given as:

1) Compute $m = c\hat{}d \; mod \; n$ by using private key.

2) Extracts the plain text from integer representing m.

3.3 The security of the RSA cryptosystem:

The security of RSA depends upon many factors that is value of p, q, e and d. Various problems persist if these parameters are not taken properly and some of them are discussed below [40].

3.3.1 The factoring problem:

If hacker wants to decrypt a cipher text message c into the plain text message m knowing only the public key (n, e) then such an opponent must try to factor n in order to determine z.

Factorization of a prime by using Euler's Factorization Method

It is a technique for factoring a number by writing it as a sum of squares in two different ways. Suppose that the number to be factored is n.

$$n = a^2 + b^2 = c^2 + d^2 \tag{3.1}$$

First deduce that

$a^2 - c^2 = d^2 - b^2$

$(a - c)(a + c) = (d - b)(d + b)$

Now let $k = gcd\,(a - c, d - b)$

And $h = gcd\,(a + c, d + b)$

So when $gcd\,(l, m) = 1$ then we say that

$(a - c) = kl$ (3.2)

$(d - b) = km$ (3.3)

And when $gcd\,(l', m') = 1$ then we say that

$(a + c) = hm'$ (3.4)

$(d + b) = hl'$ (3.5)

Substituting these into equation (3.1) gives

$klhm' = kmhl'$

$lm' = l'm$

(l, m) and (l', m') are pairs of relatively prime numbers

$l = l'$

$m = m'$

So

$(a - c) = kl$

$(d - b) = km$

$(a + c) = hm$

$(d + b) = hl$

Then $m = gcd\,(a + c, d - b)$

$l = gcd\,(a - c, d + b)$

Applying Brahmagupta-Fibonacci identity we get

$$(k^2 + h^2)(l^2 + m^2) = (kl - hm)^2 + (kl + hm)^2 \quad (3.6)$$

$$= ((a - c) - (a + c))^2 + ((d - b) + (d + b))^2$$

$$= (2c)^2 + (2d)^2$$

$$= 4n$$

$$(k^2 + h^2)(l^2 + m^2) = (kl + hm)^2 + (km - hl)^2 \quad (3.7)$$

$$= ((a - c) + (a + c))^2 + ((d - b) - (d + b))^2$$

$$= (2a)^2 + (2b)^2$$

$$= 4n$$

So from this we get

$$n = ((k/2)^2 + (h/2)^2)\,(l^2 + m^2) \quad (3.8)$$

Example

Suppose n = 1000009

Then $1000009 = 1000^2 + 3^2 = 972^2 + 235^2$

$a = 1000$, $b = 3$, $c = 972$, $d = 235$

$a - c = 28$

$a + c = 1972$

$d - b = 232$

$d + b = 238$

$k = gcd\,(a - c, d - b) = 4$

$h = gcd\,(a + c, d + b) = 34$

$m = gcd\,(a + c, d - b) = 7$

$l = gcd\,(a - c, d + b) = 58$

Then

$1000009 = [(4/2)^2 + (34/2)^2]\,(7^2 + 58^2)$

$= 293.\ 3413$

Hence factors of 1000009 are 293 and 3413.

3.3.2 Small encryption exponent:

In order to speed up encryption process by reducing amount of operations involved one can choose a small encryption exponent e which is used for each encryption [11]. However, let us consider the case in which the same plaintext message m is encrypted and sent to k destinations using the same encryption exponent e but k distinct moduli n_i where $i = 1\ldots..k$, $k >= e$. A hacker intercepting e messages out of k sent messages was faced with the following system:

$$x \equiv c_1\ (mod\ n_1)$$

……..

……..

$$x \equiv c_e\ (mod\ n_e)$$

Where $c_i = m^e\ (mod\ n_i)$, where $i = 1\ldots..e$

In most of the real-life situations $gcd\,(n_i, n_j)$ = 1, where, j = $1\ldots..e$ and $i < j$. By applying Chinese Remainder Theorem [37] one can obtain a solution $x\ mod\ N$, with $N = \Sigma_{i=1}^{\ e}\ n_i$. Since $me < N$ and as such $x = me$ and one can reach the plaintext

message m simply by extracting the integer e-th order root of x. Avoid from using small encryptions exponents, like e = 3, in favor of larger ones, an usual choice being $e = 2^{16} + 1 = 65537$

Brute Force Search Attack

For example, consider the problem of finding an integer x such that

$x \equiv 2 \ (mod\ 3)$

$x \equiv 3 \ (mod\ 4)$

$x \equiv 1 \ (mod\ 5)$

A brute-force approach converts these congruences into sets and writes the elements out to the product of 3×4×5 = 60.

$x \in$ {2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53, 56, 59, …}

$x \in$ {3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, …}

$x \in$ {1, 6, 11, 16, 21, 26, 31, 36, 41, 46, 51, 56, …}

To find an x intersect the three sets:

$x \in$ {11, …}

This can be expressed as

x = 11 (mod 60)

Example of Chinese remainder theorem:

$x \equiv 3 \ (mod\ 7)$

$x \equiv 2 \ (mod\ 9)$

$x \equiv 8 \ (mod\ 13)$

At first, we notice that gcd (7, 9) = 1, gcd (7, 13) = 1, gcd (9, 13) = 1, therefore, the system can determine:

$N = 7 * 9 * 13 = 819$

$N_1 = 819/7 = 117$

$N_2 = 819/9 = 91$

$N_3 = 819/13 = 63$

Ki for i = 1….3:

$K_1 \equiv 117^{-1} \bmod 7 \equiv 5^{-1} \bmod 7 \equiv 3 \bmod 7$

$K_2 \equiv 91^{-1} \bmod 9 \equiv 1^{-1} \bmod 9 \equiv 1 \bmod 9$

$K_3 \equiv 63^{-1} \bmod 13 \equiv 11^{-1} \bmod 13 \equiv 6 \bmod 13$

$x \equiv [(3* 117* 3) + (2* 91* 1) + (8* 63* 6)] \bmod 819$

$x \equiv (1053 + 182 + 3024) \bmod 819$

$x \equiv 164 \bmod 819$

3.3.3 Small decryption exponent:

If the size of the decryption exponent d is less than ¼ of size of modulus N, there exist a Wiener Attack through which through which the decryption exponent d can be determined knowing only the public key [32].

Wiener's Attack

It is applicable only when $q < p < 2q$ and $d < 1/3\ N^{1/4}$

Since

$$ed = 1\ (mod\ lcm(p-1, q-1)) \tag{3.9}$$

there exists an integer K such that

$$ed = K * lcm(p-1, q-1) + 1 \tag{3.10}$$

$ed - kz = 1$

$$|e/z - k/d| = 1/dz \tag{3.11}$$

Hence k/d is an approximation of e/d. Although the attacker does not know z he may use N to approximate it. Then

$z = N - p - q + 1\ and\ p + q - 1 < 3N^{1/2}$

we have

$|p + q - 1| < 3N^{1/2}$

$$|N + 1 - z - 1| < 3N^{1/2} \tag{3.12}$$

Using N in place of z we get

$|e/N - k/d| = |(ed - kN) / Nd|$

$|e/N - k/d| = |(ed - kz - kn + kz) / Nd|$

$|e/N - k/d| = |(1 - k(N - z)) / Nd|$

$$|e/N - k/d| <= |3kN^{1/2}/ Nd| = 3k/dN^{1/2} \tag{3.13}$$

Hence $kz = ed - 1 < ed$ and so $e < z$ then

$kz < ed < zd$

$k < d$

Since $k < d$ and $d < N^{1/4}/3$ then we obtain

$|e/N - k/d| <= 1/dN^{1/4}$

Since $d < N^{1/4}/3$

Then $2d < N^{1/4}$

So $1/2d > 1/N^{1/4}$

By putting in equation (3.13) we get

$|e/N - k/d| <= 3k/dN^{1/2} < 1/d.2d = 1/2d^2$ (3.14)

3.3.4 Common modulus attack:

If a trusted authority designates a single public key modulus n and one distinct encryption or decryption exponent pair (e_i, d_i) for each entity in a network then the following situations may arise:

- any entity in the network can easily factor n knowing the private key d_t for a public-key pair (n, e_t) allows the user to easily factor n and then to determine each private key d_t in the network [33].
- an opponent outside the network who intercepts the message m being encrypted which is sent to at least two different entities, has a chance to decrypt the message as follows:

  $c_1 = m^{e1}(mod\ n)$

  $c_2 = m^{e2}\ (mod\ n)$

If $gcd\ (e_1, e_2) = 1$ then using the corollary to the extended Euclidean theorem, we get

$1 = ae_1 + be_2$ for some $a, b \in z$ (3.15)

Hence, we have:

$m \equiv m^1 \equiv m^{ae1+be2} \equiv (m^{e1})^a . (m^{e2})^b \equiv {c_1}^a . {c_2}^b\ (mod\ n)$ (3.16)

Because with a high probability $gcd\ (e_1 , e_2) = 1$. A hacker was almost able to decrypt the plaintext m without requiring knowledge of the private keys d_i. Both the situations can be avoided by providing each entity with a unique modulus in the network [23].

3.3.5 Timing attacks:

If the attacker has complete of the hardware of coder and is able to measure the encryption and decryption times for several known cipher texts than he/she can deduce the decryption key d quickly. One way to stop this attack is to ensure that decryption

operation takes a constant amount of time for every cipher text but this approach can significantly reduce the performance. The alternate way to avoid this attack is with the help of a technique known as cryptographic blinding. RSA blinding makes use of the multiplicative property of RSA. Instead of computing $c^d(mod\ n)$ one can first choose a random value r and then computes $(r^e c)^d\ (mod\ n)$ [39]. Result of this computation after applying Euler's Theorem is $rc^d\ (mod\ n)$ and effect of r can be removed by multiplying with its inverse. A new value of r was chosen for every cipher text. With blinding property the decryption time was no longer correlated to the value of the input cipher text and hence the timing attack fails.

3.3.6 Chosen Cipher Text Attack:

The RSA algorithm is exposed to a chosen cipher text attack (CCA). It is described as an attack in which opponent picks up a number of cipher texts, and then given the equivalent plain texts which are decrypted with target's private key. It offers the opponent with no new information [15]. Alternatively the adversary take advantage of RSA. It opts for blocks of data which are processed using target's private key in order to yield information required for cryptanalysis.

3.4 Proposed Algorithm:

In this algorithm, we use four prime numbers instead of two so that it become very difficult to factor n. This increases the security of the system and also increases the computation complexity. To decrease the complexity we improved the time taken by modulo operation. Both encryption and decryption in RSA involve raising an integer to another integer mod n. Since the integers are large numbers, if the exponentiation is done before modulo operation the size of the intermediate result would be very large. To make it practical to implement the RSA algorithm the following property of modular arithmetic is exploited.

$(a\ mod\ n) * (b\ mod\ n) = (a * b)\ mod\ n$

Using this property along with successive multiplication scheme it is possible to compute x^6 with less than $(e-1)$ multiplications. For example, x^{64} can be computed by computing the following intermediate results: x, x^2, x^4, x^8, x^{16}, x^{32}, x^{64}. Here the result could be obtained in 5 multiplications instead of 61.

3.5 Block Diagram of Proposed Algorithm:

In this algorithm first we have to generate four distinct prime numbers p, q, r and s. The product of these numbers is n which is a component of public key. We then generate the encryption key e for converting plain text to cipher text which must be relatively prime number to $z = (p-1) * (q-1) * (r-1) * (s-1)$. After this we generate the decryption key d for converting cipher text to plain text such that $d * e \, mod \, z$ = 1. Hence the public key is (n, e) and private key is (n, d). Then for any given plain text cipher text can be calculated and vice versa as follows:

Cipher Text = power (Plain Text, e) mod n.

Plain Text = power (Cipher Text, d) mod n.

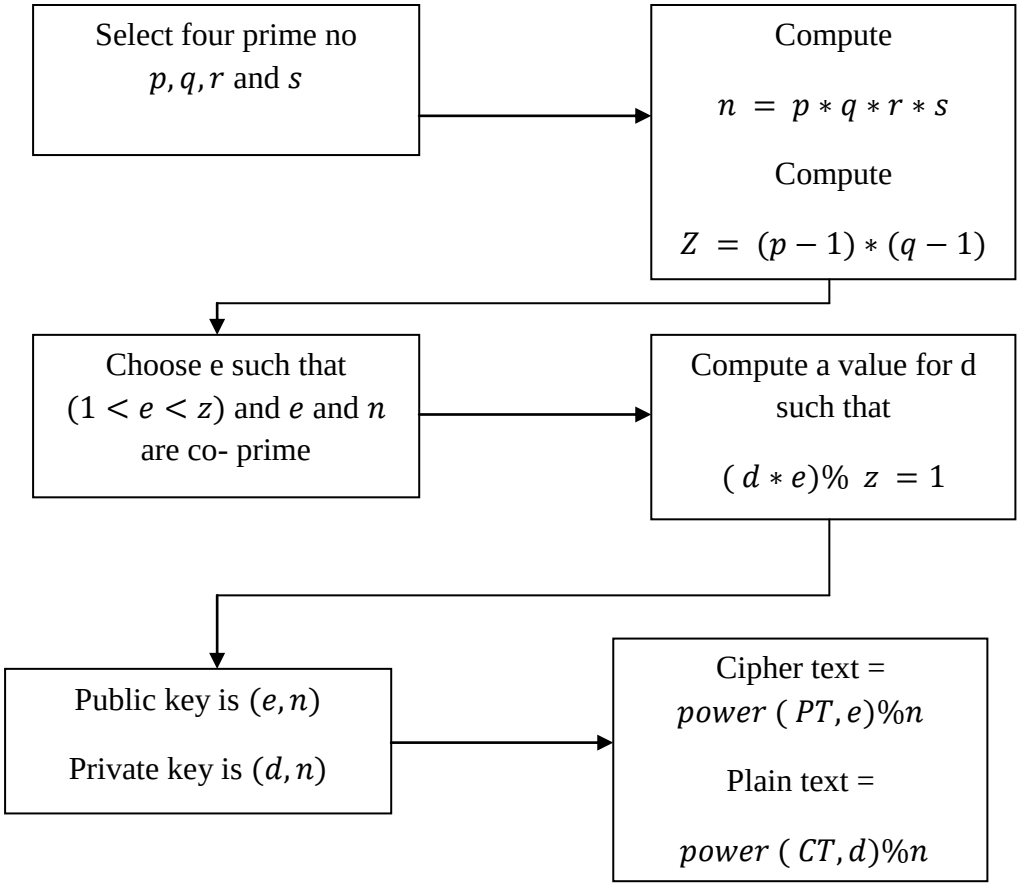


Figure 3.1: Block Diagram of Proposed Algorithm

3.6 Flowchart of Proposed Algorithm:

The flowchart of the proposed algorithm is drawn below. It gives us the idea to implement the algorithm.

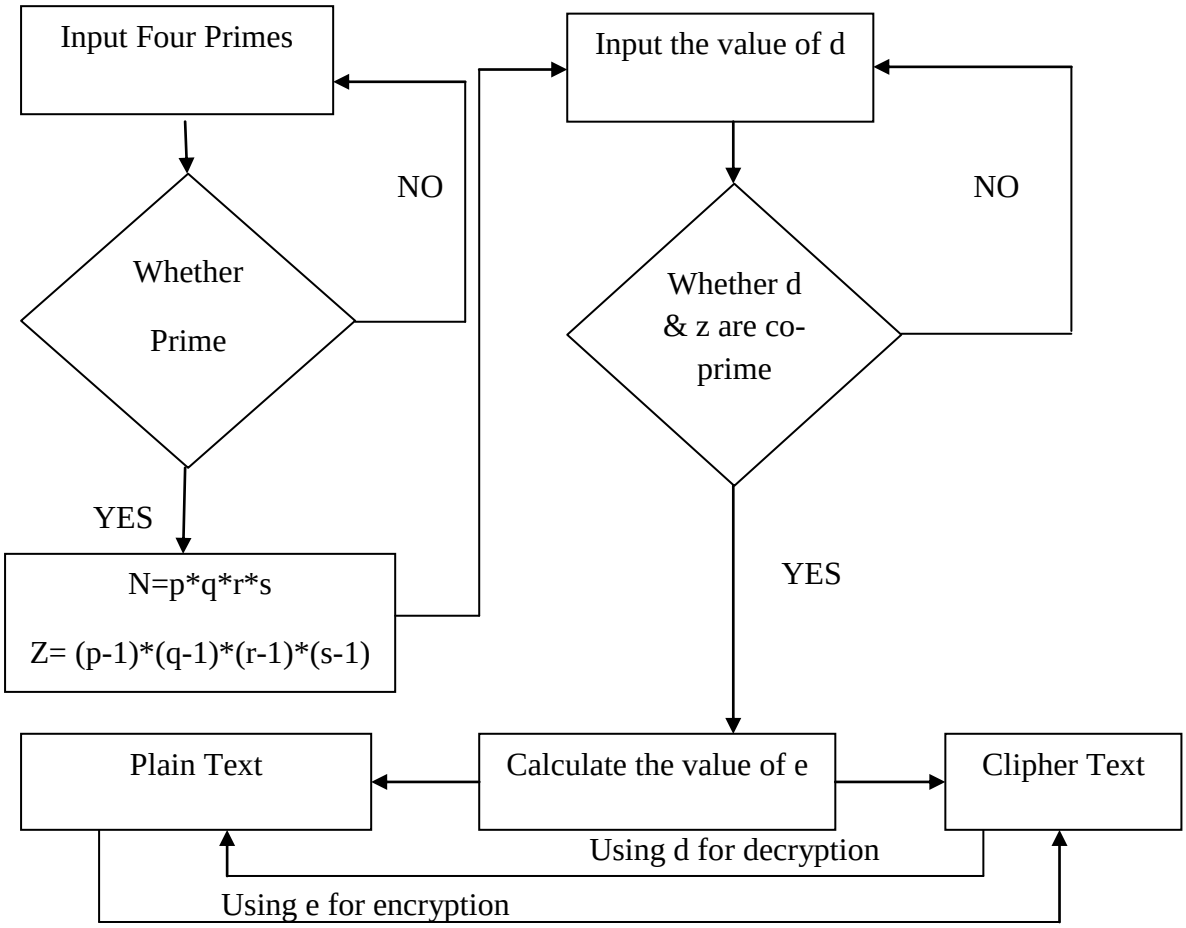


Figure 3.2: Flowchart of Proposed Algorithm

3.7 Things to be noticed while choosing N:

We know that if n can be broken down into factors with the public key $\{e, n\}$, then $z = (p-1)(q-1)(r-1)(s-1)$ cannot be hidden which makes the decoding key d a secret no more and the whole system becomes insecure. As a result, it is critical to choose the public key n while applying proposed algorithm. So, we have two choose strong prime numbers so that nobody is able to get p, q, r and s from n. If prime number p can satisfy the following requirements, then this prime number is called a strong prime number: 1) Two large prime numbers $p1$ and $p2$. 2) Four large prime numbers $r1$, $s1$, $r2$ and $s2$. Strong primes are stated as follows. Prime numbers such as $r1$ $s1$, , $r2$ and $s2$ are termed as level-3 primes and $p1$ and $p2$ are called level-2 primes. As for p, it is named as

level-1 prime. Obviously, general prime numbers belong to level-3 primes whereas strong primes belong to level-1 category.

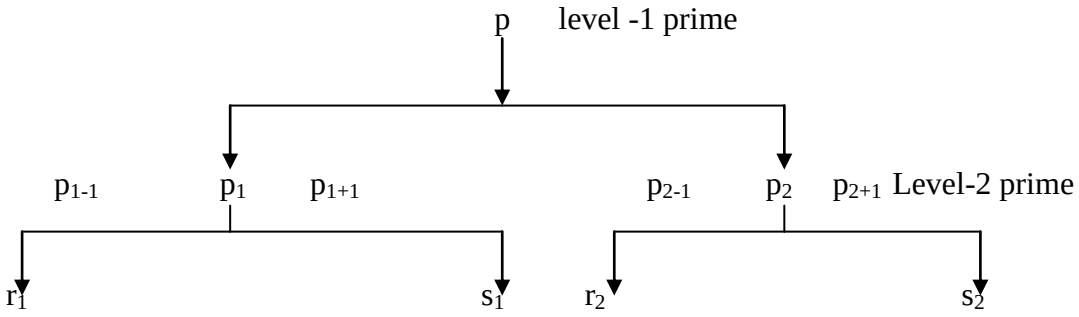


Figure 3.3: Structure of Strong Prime

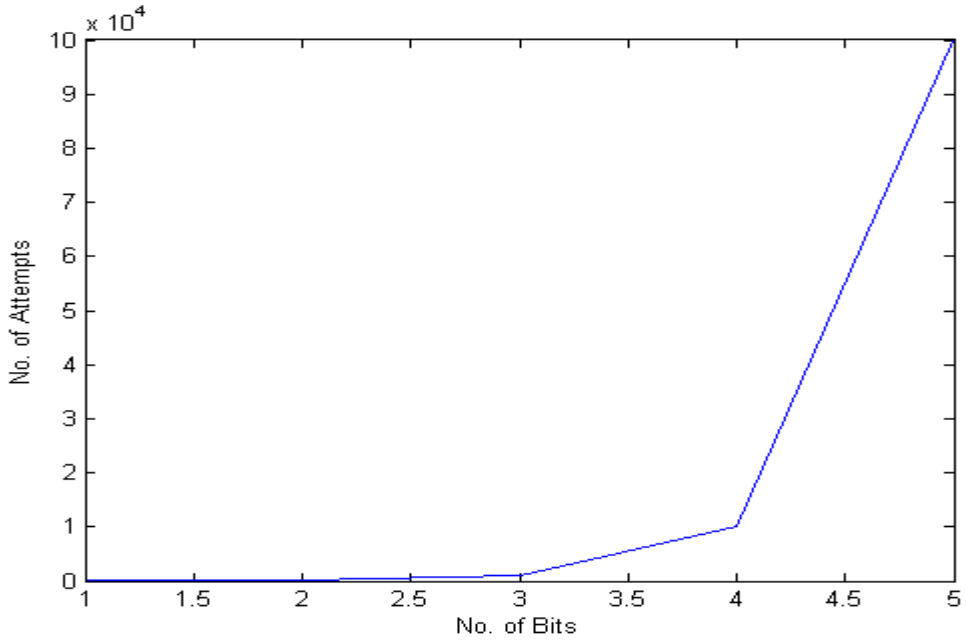
3.8 Effect of Key Size variation on proposed algorithm:

The table shown below shows that if the length of the key bit is 1 then we will recover the key surely after the 10 attempts. If the length of the key is increased to 5 then 100000 attempts are required which takes a lot of time recover the key. This means that security keeps on increasing as the key size increases but complexity of the system is also increased. So there is a need to maintain tradeoff between key length and complexity of the system.

Table 3.1: Bit Length Effect of Key on Security

S. No.	Key Length (Bits)	Number of Attempts
1	1	10
2	2	100
3	3	1000
4	4	10000
5	5	100000

The curve shows that if the number of bits of key length is increased than the security of system increases. The system is safe for 10000 attempts in case of 4bit, during this time information is reached at the destination and at the second time transmission the key is changed. If the time is less and not provide full satisfaction of safety, use the 5 bit key length. Depending upon the requirements key length is increased.



Graph 3.1: Bit Length Effect of Key on Security

3.9 Outcome:

After studying RSA it is found that there are many flaws in it. To overcome these flaws optimization has been done in the algorithm which uses four primes instead of two. It increases the complexity of the system and to overcome this, dynamic keys have been used to improve the time taken by modulo operation.

CHAPTER 4: Comparison between Proposed Approach and Other Algorithms

In this chapter comparison between proposed approach and other algorithms has been done on the basis of throughput for different file sizes.

4.1 Performance Evaluation Parameters:

Performance measurement criteria are time taken by the algorithms to perform the encryption and decryption of the input text file that is encryption computation time and decryption computation time.

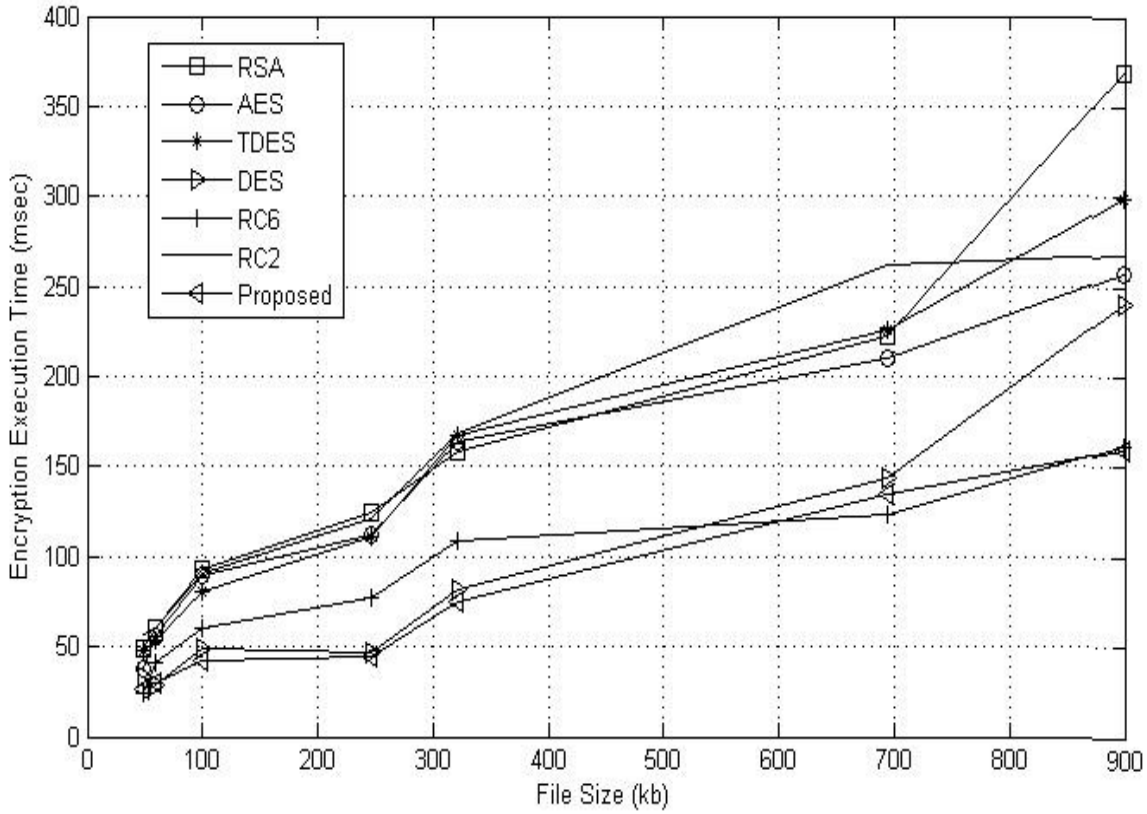
4.1.1 Encryption Computation Time: The encryption computation time is the time which is taken by the algorithms to produce the cipher text from the plain text. The encryption time can be used to calculate the encryption throughput of the algorithms.

Table 4.1: Encryption Execution Time for Different File Sizes

Input File Size(Kb)	Encryption Execution Time(msec)						
	RSA	**AES**	**TDES**	**DES**	**RC6**	**RC2**	**Proposed**
49	49	38	48	33	24	57	27
59	60	56	54	29	41	60	30
100	93	90	81	49	60	91	42
247	125	112	111	47	77	121	45
321	158	164	167	82	109	168	75
694	222	210	226	144	123	262	135
899	369	258	299	240	162	268	160
5345.28	1441	1237	1466	1296	695	1570	660

For the file of 49Kb in size the encryption execution time for RSA, AES, TDES, DES, RC6, RC2 and proposed algorithm are 49, 38, 48, 33, 24, 57 and 27msec respectively. For file size of 5345.28Kb the encryption execution time are 1441, 1237, 1466, 1296, 695, 1570 and 660msec respectively. Since, it has been shown that proposed algorithm consumes less time for all types of file sizes. With the help of the table 4.1 graph has been

drawn and it shows that how the encryption execution time depends on the file size. Finally comparison of proposed approach with various algorithms has been done.



Graph 4.1: Encryption Execution Time for Different File Sizes

Calculation of Encryption Throughput:

Encryption Throughput (Kb/sec) = Σ Input File Size/Σ Encryption Execution Time

Σ Input file Size = 49 + 59 + 100 + 247 + 321 + 694 + 899 + 5345.28

Σ Input file Size = 7714.28 Kb.

Encryption Throughput for RSA:

Σ Encryption Execution Time [RSA] = 49+60+93+125+158+222+369+1441

Σ Encryption Execution Time [RSA] = 2517

Encryption Throughput [RSA] =7714.28/2517

Encryption Throughput [RSA] = 3.06 Kb/msec.

Encryption Throughput for AES:

Σ Encryption Execution Time [AES] =38+56+90+112+164+210+258+1237

Σ Encryption Execution Time [AES] = 2165

Encryption Throughput [AES] = 7714.28/2165

Encryption Throughput [AES] = 3.56 Kb/msec.

Encryption Throughput for TDES:

Σ Encryption Execution Time [TDES] =48+54+81+111+167+226+299+1466

Σ Encryption Execution Time [TDES] = 2452

Encryption Throughput [TDES] = 7714.28/2452

Encryption Throughput [TDES] = 3.14 Kb/msec.

Encryption Throughput for DES:

Σ Encryption Execution Time [DES] = 29+33+49+47+82+144+240+1296

Σ Encryption Execution Time [DES] = 1920

Encryption Throughput [DES] = 7714.28/1920

Encryption Throughput [DES] = 4.01 Kb/msec.

Encryption Throughput for RC6:

Σ Encryption Execution Time [RC6] =24+41+60+77+109+123+162+695

Σ Encryption Execution Time [RC6] = 1291

Encryption Throughput [RC6] = 7714.28/1291

Encryption Throughput [RC6] = 5.97 Kb/msec.

Encryption Throughput for RC2:

Σ Encryption Execution Time [RC2] =57+60+91+121+168+262+268+1570

Σ Encryption Execution Time [RC2] = 2597

Encryption Throughput [RC2] = 7714.28/2597

Encryption Throughput [RC2] = 2.97 Kb/msec.

Encryption Throughput for Proposed Algorithm:

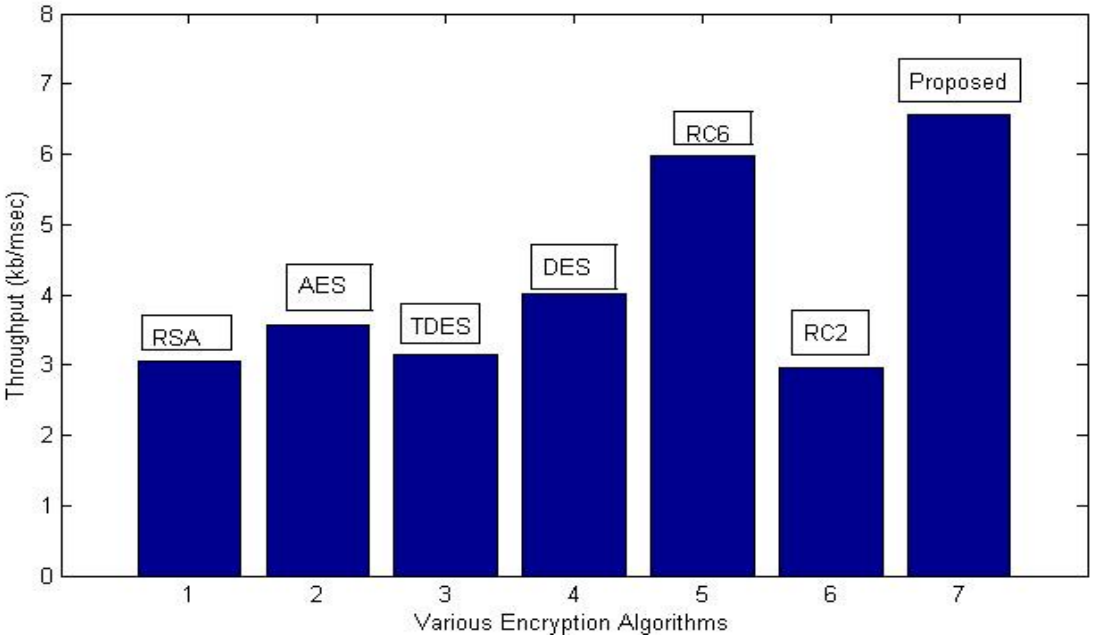
Σ Encryption Execution Time [Proposed] =27+30+42+45+75+135+160+660

Σ Encryption Execution Time [Proposed] = 1174

Encryption Throughput [Proposed] = 7714.28/1174

Encryption Throughput [Proposed] = 6.57 Kb/msec.

From the above calculated values of throughput; it is clear that proposed algorithm provides optimized results in comparison to other encryption algorithms and shown in graph 4.2.



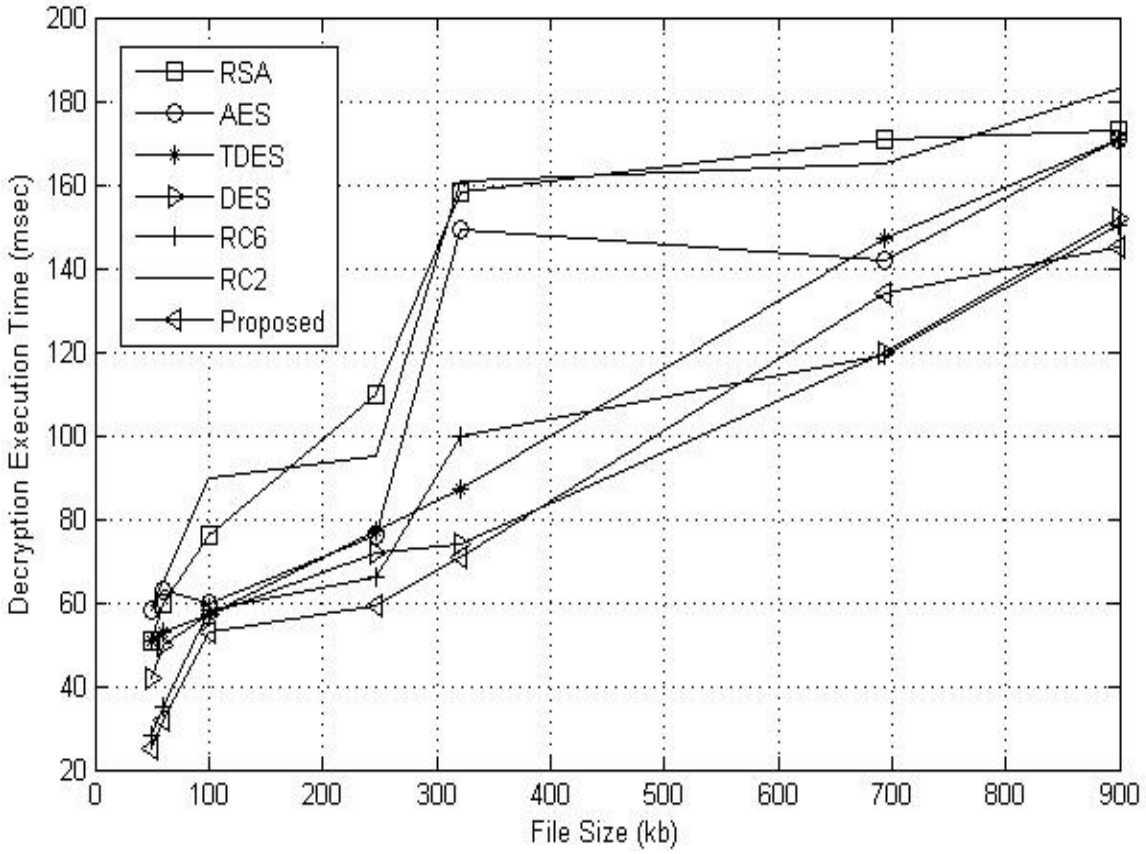
Graph 4.2: Throughput of Various Encryption Algorithms

4.1.2 Decryption Computation Time: The decryption computation time is the time taken by the algorithms to produce the plain text from the cipher text. The decryption time can be used to calculate the decryption throughput of the algorithms.

Table 4.2: Decryption Execution Time Table for Different File Sizes

Input File Size(Kb)	**Decryption Execution Time(msec)**						
	RSA	**AES**	**TDES**	**DES**	**RC6**	**RC2**	**Proposed**
49	51	58	51	42	28	59	25
59	60	63	53	50	35	65	32
100	76	60	57	57	58	90	53
247	110	76	77	72	66	95	59
321	158	149	87	74	100	161	71
694	171	142	147	120	119	165	134
899	173	171	171	152	150	183	145
5345.28	880	655	835	783	684	1216	619

For the file of 49Kb in size the decryption execution time for RSA, AES, TDES, DES, RC6, RC2 and proposed algorithm are 51, 58, 51, 42, 28, 59 and 25msec respectively. For file size of 5345.28Kb the decryption execution time are 880, 655, 835, 783, 684, 1216 and 619msec respectively. Since, it has been shown that proposed algorithm consumes less time for all types of file sizes. With the help of the table 4.2 graph has been drawn and it shows that how the decryption execution time depends on the file size. Finally comparison of proposed approach with various algorithms has been done.



Graph 4.3: Decryption Execution Time Table for Different File Sizes

Calculation of Decryption Throughput:

Decryption Throughput (Kb/sec) = Σ Input File Size/Σ Decryption Execution Time

Σ Input file Size = 49 + 59 + 100 + 247 + 321 + 694 + 899 + 5345.28

Σ Input file Size = 7714.28 Kb.

Decryption Throughput for RSA:

Σ Decryption Execution Time [RSA] = 51+60+76+110+158+171+173+880

Σ Decryption Execution Time [RSA] = 1679

Decryption Throughput [RSA] =7714.28/1679

Decryption Throughput [RSA] = 4.59 Kb/msec.

Decryption Throughput for AES:

Σ Decryption Execution Time [AES] =58+63+60+76+149+142+171+655

Σ Decryption Execution Time [AES] = 1374

Decryption Throughput [AES] = 7714.28/1374

Decryption Throughput [AES] = 5.61 Kb/msec.

Decryption Throughput for TDES:

Σ Decryption Execution Time [TDES] =51+53+57+77+87+147+171+835

Σ Decryption Execution Time [TDES] = 1478

Decryption Throughput [TDES] = 7714.28/1478

Decryption Throughput [TDES] = 5.22 Kb/msec.

Decryption Throughput for DES:

Σ Decryption Execution Time [DES] = 42+50+57+72+74+120+152+783

Σ Decryption Execution Time [DES] = 1350

Decryption Throughput [DES] = 7714.28/1350

Decryption Throughput [DES] = 5.71 Kb/msec.

Decryption Throughput for RC6:

Σ Decryption Execution Time [RC6] =28+35+58+66+100+119+150+684

Σ Decryption Execution Time [RC6] = 1240

Decryption Throughput [RC6] = 7714.28/1240

Decryption Throughput [RC6] = 6.22 Kb/msec.

Decryption Throughput for RC2:

Σ Decryption Execution Time [RC2] =59+65+90+95+161+165+183+1216

Σ Decryption Execution Time [RC2] = 2034

Decryption Throughput [RC2] = 7714.28/2034

Decryption Throughput [RC2] = 3.79 Kb/msec.

Decryption Throughput for Proposed Algorithm:

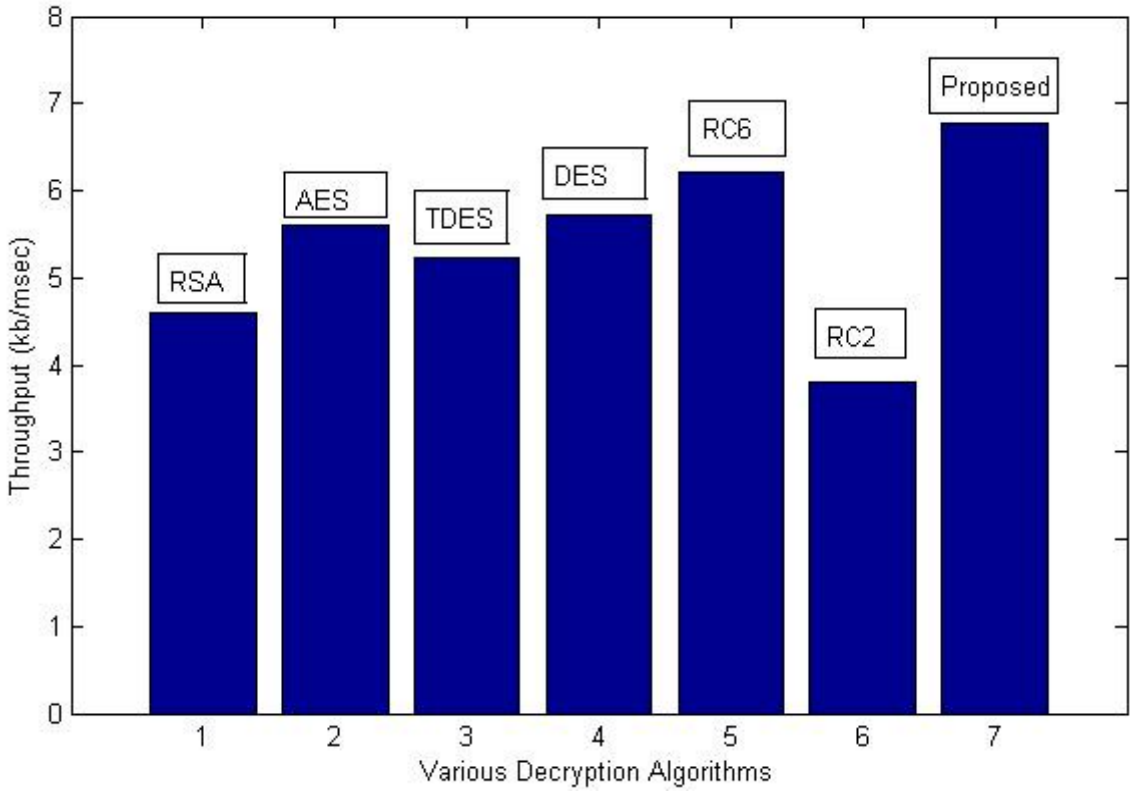
Σ Decryption Execution Time [Proposed] =25+32+53+59+71+134+145+619

Σ Decryption Execution Time [Proposed] = 1138

Decryption Throughput [Proposed] = 7714.28/1138

Decryption Throughput [Proposed] = 6.77 Kb/msec.

From the above calculated values of throughput; it is clear that proposed algorithm provides optimized results in comparison to other decryption algorithms and shown in graph 4.4.



Graph 4.4: Throughput of Various Decryption Algorithms

4.2 Outcome:

It has been observed that proposed approach provides better throughput for all types of file sizes when compared to other algorithms. Results prove that the proposed algorithm is optimized compared to other algorithms in terms of hacking and processing time.

CHAPTER 5: Results and Discussion

Simulation results have also been drawn using MATLAB 7.3. Following example shows the transmission and reception with any length of string. To implement proposed algorithm we have to focus on three parts which are a) key generation, b) encryption process, and c) decryption process.

5.1 Key Generation:

Generate four large prime numbers p, q, r and s. Here first we have to input four large prime numbers and then we calculate the value of e and d which were used to generate private and public key respectively.

Choose p = 113, q = 17, r = 211, s=97

Compute $n = p * q * r *$ s = 39317107

Compute $z = (p - 1) * (q - 1) * (r - 1) * (s - 1)$ = 36126720

Choose a number relatively prime to z and call it d.

Let d = 11

Find e such that $e * d = 1 \ mod \ z$.

e =22989731

Public key is (e, n) => (22989731, 39317107)

Private Key is (d, n) => (11, 39317107)

5.2 Encryption Process:

With the help of public key we are able to encrypt the value of plain text. Enter the value of plain text and we get the cipher text.

Enter the message you want to encrypt:

MY NAME IS HITESH MITTAL

ASCII Code of the message

M = 77

Y = 89

Space = 32

N = 78

A = 65

M = 77

E = 69

Space = 32

I = 73

S = 83

Space = 32

H = 72

I = 73

T = 84

E = 69

S = 83

H = 72

Space = 32

M = 77

I = 73

T = 84

T = 84

A = 65

L = 76

Compute the cipher text $c = m\text{^}e \ mod \ n$.

Cipher Text of the message:

M = 34620991

Y = 37669395

Space = 10441596

N = 21111725

A = 7209591

M = 34620991

E = 9042165

Space = 10441596

I = 16121978

S = 30139683

Space = 10441596

H = 32242706

I = 16121978

T = 15061200

E = 9042165

S = 30139683

H = 32242706

Space = 10441596

M = 34620991

I = 16121978

T = 15061200

T = 15061200

A = 7209591

L = 655760

5.3 Decryption Process:

With the help of the private key the cipher text can be converted to plain text.

Compute $m = c\hat{}d \ mod \ n$ by using private key.

Decrypted value of the cipher text

34620991 = M

37669395 = Y

10441596 = Space

21111725 = N

7209591 = A

34620991 = M

9042165 = E

10441596 = Space

16121978 = I

30139683 = S

10441596 = Space

32242706 = H

16121978 = I

15061200 = T

9042165 = E

30139683 = S

32242706 = H

10441596 = Space

34620991 = M

16121978 = I

15061200 = T

15061200 = T

7209591 = A

655760 = L

The decoded message is:

MY NAME IS HITESH MITTAL.

5.4 Outcome:

When the cipher text is decrypted with the help of private key, same plain text has been observed. This shows that the accuracy of proposed algorithm is very good.

CHAPTER 6: Conclusion and Future Scope

The aim of the cryptography is to prevent data from hackers. Study of various encryption algorithms has been successfully done. It has been shown that the strength of the algorithm depends on the length of the key. Key length is directly proportional to security and inversely proportional to performance. As the key length is increased the security of algorithm is also increased but performance degrades and vice-versa; key length has been optimized. After critically analyzing RSA; it is found that there are some flaws in it and to overcome these flaws a new algorithm has been proposed. The proposed algorithm increases the security of the system and also reduces the computation time; therefore hacking time is reduced which indicate that the time available for the hacker has been reduced. The proposed algorithm has been compared with other algorithms, and it is found that throughput of proposed algorithm is greater than other encryption algorithms. The work can be extended to decrease the complexity of proposed algorithm.

References:

[1] A. J. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists", IEEE Transactions on Very Large Scale Integration Systems, Vol. 9, No. 4, pp. 545-557, 2001.

[2] A. Khalique, K. Singh and S. Sood, " A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards", International Journal of Computer Applications, Vol. 2, No.3, pp. 26-30, 2010.

[3] A. Kakkar and P. K. Bansal, "Reliable Encryption Algorithm used for Communication", M. E. Thesis, Thapar University, 2004.

[4] Bharat B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan and K.S. Trivedi, "A Method for Modeling and Quantifying the Security Attributes of Intrusion Tolerant Systems", Journal of Performance Evaluation, Elsevier Science Publishers, Vol. 56, No. 1, pp. 167-186, 2004.

[5] C. H. Kim, "Improved Differential Fault Analysis on AES Key Schedule", IEEE Transactions on Information Forensics and Security, Vol. 7, No. 1, pp. 41-50, 2012.

[6] C. P. Su, T. F. Lin, C. T. Huang and C. W. Wu, "A High-Throughput Low Cost AES Processor", IEEE Communications Magazine, Vol. 41, No. 12, pp. 86-91, 2003.

[7] E. Bertino, N. Shang and S. S. Wagstaff, "An Efficient Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting", IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 2, pp. 65-70, 2008.

[8] F. Menichelli, R. Menicocci, M. Olivieri and Alessandro Trifiletti, "High Level Side Channel Modeling and Simulation for Security Critical Systems on Chips", IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 3, pp. 164-175, 2008.

[9] H. Chan, A. Perrig and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks", Proceedings of the IEEE Symposium on Security and Privacy- SP '03, California, USA, 11-14 May, 2003, pp. 197-213.

[10] H. C. Williams, "A Modification of the RSA Public-Key Encryption Procedure", IEEE Transactions on Information Theory, Vol. 26, No. 6, pp. 726-729, 1980.

[11] Hua Li and J. Li, "A New Compact Dual-Core Architecture for AES Encryption and Decryption", IEEE Canadian Journal of Electrical and Computer Engineering, Vol. 33, No. 3, pp. 209-213, 2008.

[12] H. Chien, “Efficient Time-Bound Hierarchical Key Assignment Scheme”, IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, pp. 1301-1304, 2004.

[13] H. W. Kim and S. Lee, “Design and Implementation of a Private and Public Key Crypto Processor and Its Application to a Security System”, IEEE Transactions on Consumer Electronics, Vol. 50, No. 1, pp. 214-224, 2004.

[14] H. Yang, E. Osterweil, D. Massey, S. Lu, and L. Zhang, “Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC”, IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 5, pp. 656-669, 2011.

[15] H. M. Sun, M. E. Wu, W. C. Ting, and M. J. Hinek, “Dual RSA and Its Security Analysis”, IEEE Transactions on Information Theory, Vol. 53, No. 8, pp. 2922-2933, 2007.

[16] Jason H. Li, B. Bhattacharjee, M. Yu and Levy, “A Scalable Key Management and Clustering Scheme for Wireless Adhoc and Sensor Networks”, Journal of Future Generation Computer Systems, Elsevier Science Publishers, Vol. 24, pp. 860-869, 2008.

[17] J. T. Park, J. W. Nah and W. H. Lee, “Dynamic Path Management with Resilience Constraints under Multiple Link Failures in MPLS/GMPLS Networks”, IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 3, pp. 143-154, 2008.

[18] J. Ren and L. Harn, “Generalized Ring Signatures”, IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 3, pp. 155-163, 2008.

[19] 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.

[20] L. Hundessa and J. Domingo-Pascual, “Optimal and Guaranteed Alternative LSP for Multiple Failures”, Proceedings of 13 IEEE International Conference on Computer Communication and Networks ICCCN ’04, Chicago, USA, 2004, pp. 59-64.

[21] L. J. G. 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.

[22] M. E. Hellman, “An Extension of the Shannon Theory Approach to Cryptography”, IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 289-294, 1977.

[23] M. A. Hasan, “Power Analysis Attacks and Algorithmic Approaches to their Countermeasures for Koblitz Curve Cryptosystems”, IEEE Transactions on Computers, Vol. 50, No. 10, pp. 1071-1083, 2001.

[24] M. Y. Wang, C. P. Su, C. L. Horng, C.W. Wu and C. T. 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.

[25] M. Y. Wang and C.W. Wu, “A Mesh-Structured Scalable IPsec Processor”, IEEE Transactions on Very Large Scale Integration Systems, Vol. 18, No. 5, pp. 725-731, 2010.

[26] M. Alioto, M. Poli and S. Rocchi, “Differential Power Analysis Attacks to Precharged Buses: A General Analysis for Symmetric-Key Cryptographic Algorithms”, IEEE Transactions on Dependable and Secure Computing, Vol. 7, No. 3, pp. 226-239, 2010.

[27] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu and T. L. Porta, “Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks”, IEEE Transactions on Mobile Computing, Vol. 6, No. 6, pp. 663-677, 2007.

[28] 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.

[29] S. K. Lee and D. Griffith, “Hierarchical Restoration Scheme for Multiple Failures in GMPLS Networks”, Proceedings of 31st International Conference on Parallel Processing Workshops -ICW ’02, Canada, 20-23rd August, 2002, pp. 177-182.

[30] S. T. Halkidis, Nikolaos Tsantalis and Alexander Chatzigeorgiou, “Architectural Risk Anaylisis of Software Systems Based on Security Patterns”, IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 3, pp. 129-142, 2008.

[31] S. Mangard, M. Aigner and S. Dominikus, “A Highly Regular and Scalable AES Hardware Architecture”, IEEE Transactions on Computers, Vol. 52, No. 4, pp. 483-491, 2003.

[32] S. F. Mare, M. Vladutiu and L. Prodan, "Secret data communication system using Steganography, AES and RSA", IEEE 17th International Symposium for Design and Technology in Electronic Packaging, pp. 339-344, 2011.

[33] S. Bu, F. R. Yu, X. P. Liu, P. Mason and H. Tang, "Distributed Combined Authentication and Intrusion Detection With Data Fusion in High-Security Mobile Networks", IEEE Transactions on Vehicular Technology, Vol. 60, No. 3, pp. 1025-1036, 2011.

[34] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol. 31, No. 4, pp. 469-472, 1985.

[35] T. Rams and P. Pacyna, "A Survey of Group Key Distribution Schemes With Self-Healing Property", IEEE Communications Surveys and Tutorials, Vol. 15, No. 2, pp. 820-842, 2013.

[36] T. Fukunaga and J. Takahashi, "Practical Fault Attack on a Cryptographic LSI with ISO/IEC 18033-3 Block Ciphers", Proceedings of International Workshop on Fault Diagnosis and Tolerance in Cryptography FDTC'09, Lausanne, Switzerland, 6th September 2009, Sponsored by IEEE Computer Society, pp. 84-92.

[37] W.Stallings, "Cryptography and Network Security", 2nd Edition, Prentice Hall, 1999.

[38] X. Zhang and K. K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm", IEEE Transactions on Very Large Scale Integration Systems, Vol. 12, No. 9, pp. 957-967, 2004.

[39] Y. Li, K. Ohta and K. Sakiyama, "New Fault-Based Side-Channel Attack Using Fault Sensitivity", IEEE Transactions on Information Forensics and Security, Vol. 7, No. 1, pp. 88-97, 2012.

[40] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks", IEEE Transactions Selected Areas in Communications, Vol. 24, No. 2, pp. 1-14, 2006.

[41] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing", IEEE Transactions on Information Forensics and Security, Vol. 7, No. 2, pp. 743-754, 2012.