CHAPTER - 1
INTRODUCTION

“If privacy is outlawed, only outlaws will have privacy.”
- Philip Zimmermann

1.1 History of Cryptography

As the demand for effective data security is increasing day by day, any organization has an obligation to protect secret and sensitive data from theft or loss. Such sensitive data can be potentially damaging if it is altered, destroyed, or hacked. This makes it necessary to protect the data. Cryptography attempts to provide such guarantee and it ensures the security of data being transmitted. It is implemented in many day-to-day applications such as the security of ATM card, computer passwords, e-commerce, military, etc.

The word “cryptography” comes from the Latin root “kryptos” which means hidden and “graphein” means writing. Hence, cryptography refers to secret writing. Until modern era, cryptography was exclusively referred to as encryption. It is the process of converting messages from a comprehensive form into an incomprehensible one at one end and which reverses the process at the other end so that the message is unreadable by interceptors or eavesdropper without the secret knowledge.

The earliest forms of secret writing was the first phase of cryptography and it required just little more than local pen and paper period analogs as most people could not read. In this period, the methods used for maintaining the secrecy were not so difficult, since messages were decoded by hand. The two important classical ciphers used in that period are: substitution cipher and transposition cipher. In transposition cipher, the letters in a message are rearranged so that the message “bard” would be encoded as “rdba”. In substitution cipher, a letter or groups of letters are replaced by another letter or groups of letters respectively. One of the earliest substitution ciphers was the Caesar Cipher, in which the letter in the plaintext was replaced by shifting each letter forward by fixed number of positions. It was named after Julius Caesar who invented this cipher. He used a shift by three so that the message “bharathidasan” would be
encoded as "ekdudwkmgdwdq" just like excess-3 code in Boolean algebra. The earliest known use of cryptography was found in some carved ciphertext on stone in Egypt (1900 B.C). The oldest was bakery recipes from Mesopotamia, historically [1].

The ‘mechanical era’ was the second phase of cryptography. Many encryption/decryption devices were invented early in the 20\textsuperscript{th} century and several patented. The most important was rotor machines-famously including the Enigma founded by German which was used to generate the code. The breaking of the Enigma code was an important event that determined the course of World War II and according to most experts shortened the war by about a year. The Germans figured the Enigma code was unbreakable. British code breaking organization employed over 10,000 people breaking these codes, but the level of effort was very high, the Germans had anticipated.

The third phase of the cryptography is the ‘modern era’ [2]. In this era, the development of digital computers and electronics after World War II made possible much more complex ciphers. Computers allowed for the encryption of any kind of data representable in any binary format, but classical ciphers used to encrypt only written language texts which were the new and remarkable achievement in cryptography. Extensive research was done in the early 1970s. Some of the famous algorithms were: DSA; Whitefield Diffie and Martin Hellman key exchange algorithm, RSA etc. Since then, cryptography has been used as a tool in communications, information security etc.

Some modern cryptographic algorithms can keep their keys secret, if certain mathematical problems are intractable such as discrete logarithm problems, integer factorization problem, and sum of subset problems. Even though there is no absolute proof to prove the security of the cryptographic techniques, the methodology used in the techniques is very difficult to solve. Hence, more formally, cryptography is the study of mathematical techniques related to the security services. It is deployed with cryptographic algorithms for hiding message and retrieving hidden messages.

### 1.2 Background

In cryptography, the message which is to be kept secret is called as plaintext. The process of hiding its content is called encryption and the encrypted message is called ciphertext. The process of receiving the content of plaintext from the ciphertext is
A cryptographic algorithm or cipher is a mathematical function used in the encryption and decryption processes. A modern cryptographic algorithm always includes a key. Cryptographic algorithms, plaintexts, ciphertexts, and keys are collectively called cryptosystem. It works in combination with a key to encrypt the plaintext and to decrypt the ciphertext.

While cryptography is the science of secure data, cryptanalysis is the science of analyzing and breaking secure communication. An attempted cryptanalysis is called attack and its practitioner is called an attacker. The branch of mathematics which encompasses both cryptography and cryptanalysis is called cryptology and its practitioner is generally called cryptologist. A fundamental assumption in cryptanalysis was first stated by the Dutchman A. Kerkhoff. It is usually referred as Kerkhoff's principle. It states that the secrecy must reside entirely in the key even though the adversary knows the details of the cryptosystem, including algorithms and their implementations.

The fundamental and classical task of cryptography is not only providing mechanism for encryption but also to provide the security services requirements like authentication, confidentiality, integrity, and non-repudiation defined by ITU-T X.800 (International Telecommunication Union-Security Recommendations). The service confidentiality protects the content of information from unauthorized entity. Data integrity is the service which gives the assurance that data received are exactly as sent by an authorized entity.

The service authentication applies to the communicating parties as well as the information. It is divided into peer entity and data origin authentication. Peer entity authentication is used with a logic connection to provide confidence in the identity of the entities connected and data entity authentication implicitly provides data integrity. Similarly, non-repudiation is a service which prevents protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. To provide such security services, most systems use two major classes of cryptographic algorithms viz., symmetric and public-key algorithms which are shown in Figure 1.1. An overview of these two algorithms is presented in sections 1.3 and 1.4.
1.3 Symmetric-Key Algorithms

In symmetric-key algorithms [6], the sender and the receiver must agree upon a secret key prior to the communication taking place. It is also called private-key or secret-key algorithm, where the same key can be used for both encryption and decryption, or the decryption key can easily be calculated from the encryption key and vice versa.

![Diagram of Cryptographic Algorithms]

**Figure 1.1: Types of Cryptographic Algorithms**

The security of these algorithms depends on maintaining the secrecy of the key as soon as the communication is over. This was the only kind of encryption publicly known until 1976. There are two main disadvantages in this kind of cryptography. First, it is difficult to ensure the security of the keys, when a secure channel does not already exist between them. Another disadvantage is, it requires complex key management. Symmetric-key algorithms are further classified into block ciphers and stream ciphers.

1.3.1 Block Ciphers

Block ciphers [7] are encryption/decryption schemes in which the message is divided into various blocks with equal size and it encrypts one block at a time. Many block ciphers have a Feistal structure. A cryptographic system based on Feistel structure uses the same basic algorithm for both encryption and decryption. Named after the IBM cryptographer Horst Feistel and first implemented in the Lucifer cipher by Horst Feistel and Don Coppersmiths [8]. Feistal cipher consists of a number of identical rounds of processing. In each round, a substitution is performed on half of the data being processed, followed by a permutation that interchanges the two halves. The
original key is expanded so that a different key is used for each round. It utilizes the concept of a product cipher, which is the execution of two or more simple block ciphers. Under the control of a fixed key, different occurrences of a particular plaintext will always be encrypted as the same ciphertext block. Data encryptions Standard (DES), International Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES), RC4 (Rivest Cipher 4), RC5, (Rivest Cipher 5), Blowfish, etc., are some examples of these type of algorithms. To frustrate attacks from cryptanalyst, the size of the block has to be set large enough.

The drawback of block ciphers is repetition of text. For repeating text patterns, the same ciphertext is produced. Thus, the cryptanalyst can sometimes predict the pattern of the plaintext and try to break them. If he/she succeeds, there is a chance to recover a large portion of the original plaintext. To avoid it, block ciphers are used in various algorithm modes like Cipher Block Chaining (CBC), Cipher Feed Back (CFB), and Output Feed Back (OFB) etc.

1.3.2 Stream Ciphers

In stream cipher [9], the plaintext is encrypted on a bit by bit basis. It is much faster than block cipher. With a block cipher, the encryption of any particular plaintext will result the same ciphertext if the same key is used whereas in stream cipher, the transformation of the smaller plaintext will vary depending on when they are concatenated during the encryption process. Moreover, block ciphers operate on larger blocks of data but stream cipher operate on only smaller units of the plaintext usually bits.

In stream cipher, for encrypting the plaintext, the key is fed into an algorithm called the pseudorandom bit generator [10], which creates a long sequence of binary signals called “keystream”. To produce the ciphertext, this keystream is then mixed with the plaintext sequence, usually by XOR bitwise modulo 2. Stream cipher is further divided into two types: synchronous and asynchronous stream cipher. A synchronous stream cipher is one in which the keystream is generated independently of the plaintext and ciphertext. A self-synchronous or asynchronous stream cipher is one in which the keystream is generated as a function of the key and a fixed number of previous ciphertext digits.
1.4 Public-Key Algorithms

The public-key algorithms are based on the idea of separating the key used to encrypt a message from the one used to decrypt it. In these algorithms, each communicating party needs a pair of key (private-key $k_d$, public-key $k_e$) to communicate with other parties. The public-key is used for encryption and private-key is used for decryption. The concept of public-key cryptography was invented by Diffie and Hellman [11] in 1976. The main idea that they presented in was that it is easy to derive $k_e$ from $k_d$ but it would be infeasible to derive $k_d$ from $k_e$. Since 1976, numerous public algorithms have been proposed namely: Elgamal, McEllice, Rabin, Elliptic Curve Cryptography (ECC) etc. Public-key algorithms are broadly divided into three types [12] based on the following problems. They are:

i. Integer Factorization
ii. Discrete Logarithm
iii. An NP-Complete

The next subsections present an overview of each problem and its applications in public-key algorithms.

1.4.1 Integer Factorization Problem

Integer factorization problem is a hard mathematical problem and is one of the most widely used problems. The principle adopted in this problem is to decompose a large integer $n$ into a product of two prime numbers viz., $p$ and $q$. Pollard’s rho algorithm, Continued fraction, Quadratic-Seive [13] etc., are some examples of this type of problem. Inspired by the idea of public-key cryptography and trap door one way function, Ronald, Shamir, and Adleman (RSA) proposed a public-key cryptosystem called RSA which is related to the intractability of the integer factorization problem. In the algorithm, two primes $p$ and $q$ are chosen in such a way that $p < q < 2^p$ with $q - p > 2^{60}$ so that Fermat [14] factoring method cannot work.

1.4.2 Discrete Logarithm Problem

The discrete logarithm [15] forms the basis for some public-key algorithms like ElGamal, Diffie and Hellman key exchange, Digital Signature Algorithm (DSA), Elliptic Curve Cryptography (ECC). The utility of discrete logarithm in a cryptographic setting is that finding discrete logarithm is somewhat difficult, but the
inverse operation of exponentiation can be computed efficiently by using square-and-multiply method. To define discrete logarithm, let $\alpha$ and $\beta$ be non zero integers in $\mathbb{Z}_p$ and suppose $\beta = \alpha^x \mod p$. The problem of finding $x$ is called the discrete logarithm problem and it is denoted as $x=\log_\alpha \beta$. There are several algorithms available for calculating discrete algorithm viz., Shanks, The Index Calculus, The Pohlig-Hellman etc. All the algorithms depend on any property of multiplicative group.

1.4.3 NP-Complete Problem

Knapsack public-key encryption scheme is based on subset sum problem, which is an NP-complete problem [16]. The subset sum problem is stated as: given a set\{a_1, a_2, a_3,\ldots, a_n\} of positive integers, called knapsack set, and a positive integer $s$, determine whether or not there is a subset of the $a_j$ that sum to $s$. Equivalently, whether or not there exist $x_i \in \{0,1\}$, $1 \leq i \leq n$, such that

$$a_1 x_1 + a_2 x_2 + a_3 x_3 + \ldots + a_n x_n = s$$

...(1.1)

1.5 Types of Attacks on Cryptographic Algorithms

It is noted that the security of a cryptosystem must be entirely based on the keys. Attacks [17] on the secrecy of encryption schemes try to recover plaintexts from the ciphertexts, or even more drastically to recover the secret key. The possible attacks depend on the resources available of the adversary. Cryptographic attacks are designed to subvert the security of cryptographic algorithms and they are used to decrypt data without prior access to a key. They are part of cryptanalysis, which is the art of deciphering encrypted data. The next subsection presents various kinds of attacks normally encountered in cryptographic algorithms.

1.5.1 Ciphertext-Only Attack

In this type of attack, the cryptanalyst has the ciphertext of several messages and they have been encrypted using the same encryption algorithm. The job of cryptanalyst is to recover the plaintext as possible or he can deduce the key(s) which is used to encrypt and decrypt the message. Mathematically, it is represented as:

Given : $c_1 = E_k(p_1)$, $c_2 = E_k(p_2)$,\ldots, $c_i = E_k(p_i)$

Deduce : Either $p_1$, $p_2$,\ldots,$p_i$, $k_i$, or an algorithm to infer $p_{i+1}$ from $c_{i+1}$ = $E_k(p_{i+1})$
1.5.2 Known-Plaintext Attack

In this type of attack, the cryptanalyst knows the encryption algorithm and ciphertext to be deduced. Cryptanalyst role is to deduce the key(s) used to encrypt the message or an algorithm to decrypt the new message encrypted with the same key(s). Mathematically, it is represented as:

Given : $c_1 = E_k(p_1), c_2 = E_k(p_2), \ldots, c_i = E_k(p_i)$
Deduce : Either $k$, or an algorithm to infer $p_{i+1}$ from $c_{i+1} = E_k(p_{i+1})$

1.5.3 Chosen-Plaintext Attack

In this type, the cryptanalyst not only has access to the ciphertext and associated plaintext for several messages but also he/she chooses the specific plaintext blocks to encrypt which yield more information about the key. His/her job is to deduce the key(s) used to encrypt the messages or an algorithm to decrypt any new message encrypted with the same key(s). Mathematically, it is represented as:

Given : $p_1, c_1 = E_k(p_1); p_2, c_2 = E_k(p_2), \ldots, p_i, c_i = E_k(p_i)$,
where the cryptanalyst has to choose $p_1, p_2, \ldots, p_i$
Deduce : Either $k$ or an algorithm to infer $p_{i+1}$ from $c_{i+1} = E_k(p_{i+1})$

1.5.4 Chosen-Ciphertext Attack

In this attack, the cryptanalyst knows different ciphertexts to be decrypted and has access to the decrypted plaintext. His/Her job is to deduce the key. Mathematically, it is represented as:

Given : $c_1, p_1 = D_k(c_1), c_2, p_2 = E_k(c_2), \ldots, c_i, p_i = E_k(c_i)$
Deduce : $k$

1.5.5 Dictionary Attack

In this attack, every word in the dictionary is tried as a possible password for an encrypted message. A dictionary attack is generally more efficient than brute force attack. A brute force attack involves trying every possible key until an intelligible translation of the ciphertext into plaintext is done. Though various kinds of attacks are available, to measure the security level of cryptographic algorithms after incorporating the proposed mathematical models, dictionary attack is considered in this research.
1.6 Scope of the Research

In recent times, the analysis of cryptographic algorithms has been getting more attention. Enhancing the security and increasing the speed are the most challenging in cryptographic algorithms. Thus, there is a need to develop such algorithms to meet out these challenges. The security of many cryptographic systems depends upon the generation of unpredictable quantities such as the keystream in the one-time pad, the secret key in the DES algorithms, the prime p and q in the RSA encryption etc. In all these cases, the quantities generated must be sufficient in size and at random in the sense that the probability of any particular value being selected must be sufficiently small. Even if the above said parameters are taken carefully, none of the computational problems are fully secured enough.

Moreover, to enhance the security, normally the key is selected in such a way that the size of the key may be equal to the size of the plaintext (in symmetric-key algorithms) and the number of bits in the key is also large (in public-key algorithm). Thus, there are some problems in storing and distributing the key. To avoid them, instead of increasing the key size, the focus is given on plaintext with some other numerals rather than its ASCII value. Further, slow running cryptographic algorithms cause customer dissatisfaction and inconvenience. To achieve better speed and performance, optimization of the run time is very essential so that the cryptographic algorithms could be executed faster.

1.7 Objectives of the Research

The main goal of this research work is to design a framework named “GOMAKA”, which incorporates the various mathematical models to enhance the security and optimizing the run time in the existing cryptographic algorithms. In order to achieve the objectives, the following mathematical models (contributions made in this work) are adopted.

i. To develop a doubly even Magic Square (MS) model to enhance the security of any public-key cryptographic algorithm.

ii. To enhance the security in Merkle-Hellman knapsack public-key cryptographic algorithm based on Vertex Magic Total (VMT) labeling [18].
iii. To generate a keystream for stream cipher based text encryption using Primitive Pythagorean Triples (PPTs) [19].

iv. To increase the speed of cryptographic operations in public-key cryptographic algorithm based on fuzzy modular arithmetic and addition chain.

v. To use parallelism in cryptographic operations for increasing the speed of cryptographic algorithms.

The goals from (i) to (iii) target for enhancing security and the (iv) and (v) are used for increasing the operational speed. An overall view of each proposed models is presented in the subsections 1.7.1 and 1.7.2.

1.7.1 Models for Enhancing the Security

The efficiency of a cryptographic algorithm is based on its time taken for encryption or decryption and the way it produces different ciphertext from a plaintext. A deterministic public-key encryption algorithm sometimes leaks information to an adversary because the plaintext (M) is taken from a small set {M_1, M_2, M_3,...,M_r} of possible messages. When an adversary intercepts a ciphertext C, he/she can easily find out M by computing E(M_1), E(M_2),...,E(M_r) and comparing them with C. The RSA, the widely used public-key algorithm and other public-key algorithms may not guarantee that the ciphertext is fully secured.

An encryption method provides secrecy only if the ciphertexts appear sufficiently at random to the adversary i.e., it produces different ciphertexts for the same plaintext character so that the adversary cannot easily predict the ciphertext character in the next time. This observation was applied to symmetric-key encryption like Vernam’s one-time pad. To apply it to deterministic public-key algorithms like RSA, ElGamal, Rabin, McElliece cryptographic algorithms etc., the following mathematical models are proposed in this research.

(i) Proposed Add-On Security Model Based on MS

In this model, a doubly even MS of order n (MSn) is generated equivalent to ASCII/Unicode character set based on two different approaches. The proposed work uses numerals of the MS for encryption rather than the ASCII values of the plaintext
characters. It attempts to enhance the efficiency by providing add-on security to the cryptographic algorithm. This approach will increase the security due to its complexity in encryption process as it deals with the MS formation with seed number, starting number, and magic square sum that cannot be easily traced. This proposed work provides another layer of security to any public-key algorithms such as RSA, ElGamal etc.

Since this model is acting as a wrapper to a public-key algorithm, it ensures that the security is enhanced. To improve the security, the MS is rotated with different orientations and hence the numerals which occur in the particular position of the MS are changed. Then, the security level of the two public-key algorithms viz., RSA and Merkle-Hellman knapsack is tested by the incorporation of MS model using All Block Ciphers (ABC) Universal Hackman tool. Further, the model is verified for perfect secrecy using Shannon’s [20] information theory.

(ii) Enhancing Security in Public-Key Algorithms Based on VMT
It is noted that Merkle-Hellman knapsack cipher is an NP-complete problem which gets its security from the knapsack problem. The drawback of this cipher is that it uses the same super increasing sequence for decryption and the normal sequence for encryption for all blocks of message. It reduces the security of data because anyone can recover the bit pattern from the target sum, if the elements of super increasing knapsack are known.

Thus, an alternate approach to Merkle-Hellman knapsack cipher has been developed based on VMT labeling of a graph. In VMT, every vertex is labeled by numerals. For each VMT labeling (private-key), a transformed VMT labeling is generated (public-key) which forms a new set of values. Then, encryption is accomplished through this transformed VMT labeling and decryption is performed by the original VMT labeling as super increasing sequence used in knapsack cipher. Moreover for the same VMT, multiple VMT labeling and corresponding transformed VMT labeling are generated which ensure different ciphertext for the same plaintext. The generated ciphertexts are free from any predicted pattern and the cryptanalyst cannot decipher the ciphertext easily. To encrypt the plaintext character which is of the form of $2^n$ bits, selecting the super increasing knapsack sequence is somewhat tedious. Moreover, in
Merkle-Hellman knapsack cipher, to encrypt a message only one super increasing sequence is used for all blocks of message and if a character occurs more than once in the blocks, same target sum(ciphertext) will be produced at all times. The eavesdropper could possibly predict the pattern and recover the plaintext from the target sum.

Thus, instead of using same super increasing sequence, several solutions for VMT labeling of a graph are considered in this model and they are used for encrypting all blocks of message. Hence, the number of blocks of message to be encrypted at a time depends on the number of solutions generated for the same vertex sum. Further, this enhanced approach is better than other classical public key cryptographic algorithms due to the fact that it produces different ciphertext characters for the same plaintext character. To perform encryption based on VMT, two methods are considered in this research viz, (i) VMT labeling based on twin-factorization method proposed by Krishnappa et al. [21]. (ii) VMT labeling of Products of Cycles of any Length with Odd Cycle proposed by Teneza Kovárová [22].

In order to perform encryption using twin-factorization method, first VMT labeling is generated as proposed by Krishnappa et al. and then an algorithm has been proposed which reads the block size n (or b). It generates n solution of vertex magic sum and each solution is called as vertex magic sum sequence. These solutions are then used as private-key sequences and the public-key sequences are generated from the private key sequences as in knapsack cipher. Then, they are used for encryption process. Similarly, to perform encryption and decryption using VMT labeling of cycles of any length with odd cycles, first labeling of \( C_m \times C_n \) (where \( m, n \geq 3 \) and \( n \) is odd) is generated.

To locate the VMT labeling corresponding of the vertices, the researcher proposed the algorithms to traverse the vertices either vertically or horizontally in the labeled graph. The main advantage of this methodology over the Merkle-Hellman knapsack cipher is that in the later approach, the message size is based on the number of items in the knapsack and they need not be the multiple of 8 (because ASCII is 8 bits) and it is arbitrary in size. But in the proposed method, the message size is always considered in the form of \( 2^b \) bits, where \( b > 2 \) as in other block cipher algorithms like
DES, AES etc. Finally, these labelings are used for encryption instead of using super increasing sequence as used Merkle-Hellman knapsack cryptographic algorithm.

(iii) Key Generation Based on Primitive Pythagorean Triples for Stream Cipher Encryption

It is noted that key plays a vital role in increasing the security in cryptographic algorithms. It is known fact that as the size of the key is increased, security would also be strengthened. But if the size of the key is too large, sometimes it leads storage problem. In Vernam one-time stream cipher the size of the key is selected in such a way it should be equal to the size of the plaintext. As the size is increased, storing and remembering the key is the major issue. To compromise this, a keystream is generated for stream cipher algorithms using Primitive Pythagorean Triples (PPTs) in this research.

To generate a keystream, first PPTs are generated using Barning Tree. Then, based on the basic properties of PPTs, each PPT is identified in which class A, B, C, D, E, and F have been labeled. It is noted that the sequence of class of labels generated based on the properties of PPTs are not random at all times. Thus, a randomness test is needed. Though, several tests are available for randomness, runs up and down test are considered for checking the randomness. Once it is proved that the sequence of classes labels are random, a subsequence is selected from the sequence of labels called keystream. The size of the keystream is determined by the sender and the receiver and it is always taken as less than the length of the plaintext so that it reduces the number of keys to be stored and distributed. Then, Huffman encoding is used to generate the code for the plaintext characters. To improve the security, mutation is employed at various levels of the Huffman tree. Then, the keystream and the mutated code of plaintext is XORed as in one-time pad in producing the ciphertext. The keystream cannot be easily cracked due to the fact that the generation of keystream fully depends on the selection of starting PPT. This method also produces various ciphertext characters for the same plaintext character if it appears several times in a plaintext.

1.7.2 Models for Run Time Optimization

The implementation of cryptographic systems presents several requirements and challenges. It is noted that the speed of the encryption algorithms should not be lower
than the transmission rates of the communication links. Because, slow running cryptographic algorithm will cause customer dissatisfaction and inconvenience. In order to achieve fast encryption algorithms, following models have been proposed.

(i) **Parallelism**

To speed up cryptographic operations, the concept of parallelism is introduced. The simplest way to schedule parallel system is with a queue. Each job is submitted to the queue and upon reaching the head, is executed to completion while all other jobs wait. The queue can be hypothetically First in First out (FIFO), but the scheme extends to priority queues without loss of generality. One simple scheduling system is a space sharing system. It allows more than one job to be scheduled on a parallel system at one time. Each processor is devoted to one parallel process, and each job runs until completion without pre-emption. The space sharing model is vulnerable to large jobs monopolizing the system. The popular job scheduling method, namely, Gang Scheduling method combines both space sharing and time sharing. Each job is allotted a time slot and the job may execute in its time slot. Processors within each time slot are space shared. When a time quantum has expired, all jobs in the current slot are pre-empted and replaced with jobs in the next slot.

Maui [23] is a job scheduler specifically designed to optimize system utilization in policy driven, heterogeneous high performance computing (HPC) cluster (HPC) environment. The philosophy behind it is essentially schedule jobs in priority order and then backfill the available idle processors’ time with overridden Gang Scheduling technique. Maui has a two-phase scheduling algorithm. In the first phase, the high priority jobs are scheduled using advanced reservation. The Gang Scheduling algorithm with overriding technique is used in the second phase to schedule low-priority jobs between previously selected jobs. In this work, the Maui Scheduler with overridden Gang scheduling technique has been used to calculate the encryption/decryption time of the given message in simulated environment.

(ii) **Fuzzy Modular Arithmetic**

In most of the public-key cryptosystems like RSA, ElGamal, etc., modular exponentiation plays a vital role for performing encryption/decryption operations. The performance and practicality of such cryptosystems is primarily determined by the
implementation efficiency of the modular exponentiation. Fast exponentiation is becoming increasingly important with the widening use of encryption. In order to improve the time requirements of the encryption process, minimizing the number of modular multiplication is essential. In other public-key cryptosystems like ECC [24], scalar point multiplication, kP where k is an arbitrary integer in the range 1 < k < ord(P) and P a point in the elliptic curve is the central operation.

In cryptographic algorithms, exponent is always an integer and can be performed faster than the traditional square and multiply method by iteratively reducing the small gain if the numbers of multiplications are organized properly. For that addition chain is used. Similarly, in the case of ECC to speed up kP, fuzzy modular arithmetic has been used. Fuzzy Nodular arithmetic was proposed by Wael Adi. The key idea used in fuzzy modular arithmetic is not to compute the result exactly as in the traditional modular arithmetic because, the traditional modular arithmetic uses division operation for modulo reduction m whereas in fuzzy, the division is replaced by repeated subtraction. Thus, instead of the full reduction, a pseudo fuzzy randomized partial reduction is performed.

After developing the models for the above said concepts, they are stored as components in BCCC of GOMAKA framework. They are incorporated in the cryptographic algorithms and the time taken for encryption and decryption operations are analyzed.

1.8 Chapter Organization

The thesis is separated into twelve chapters in which the enhancing security and increasing the computational efficiency of encryption and decryption operations in public-key and symmetric-key cryptosystem is discussed. Chapter 2 provides the review of related works which enhance the security in public-key and symmetric-key cryptographic algorithms. It highlights various symmetric-key algorithms which generate the secret-key which plays a vital role for encryption and decryption process. In chapter 2, a thorough review of literature is also made relating to various techniques which increase the speed of encryption and decryption time. The reason for selecting the problem and the statement of the problem is clearly presented in chapter 3.
In order to achieve the goals, a framework named as “GOMAKA” has been developed. Various packages like application, io, maths, util, codec, and their purposes, the inbuilt functions which are available in Block Ciphers Cryptographic Class (BCCC) of codec package of the proposed framework are thoroughly discussed in chapter 4. Various models have been developed to fulfill the objectives of the research work. The models are stored as methods in BCCC so that if any user wants similar type of models, then he/she can use the methods from the BCCC of the proposed framework. The proposed mathematical models viz., MS for enhancing security, existing VMT labeling which is then used for enhancing security in public-key cryptographic algorithms, and the key generation based on PPTs for stream cipher encryption are elaborately discussed in Chapters 5, 6, and 7 respectively. For each proposed model, a numerical illustration is also presented.

Modular exponentiation is an important operation in modern cryptography. Though the exponentiation operation is performed by repeated multiplication, it is a time consuming process. To reduce the number of multiplications which involve in exponent, an addition chain is normally used. In this work, a division based approach named as “DM” has been proposed for generating the shortest addition chain. The concept of division based addition chain and its application to public-key cryptographic algorithms is presented in chapter 8.

It is noted that ECC is one of the public-key cryptography which is superior to the well-known RSA cryptography, but, it gives more security than the RSA with the equivalent key size. The central operation of ECC is scalar point multiplication, kP where k is an integer and P is a point on EC. To perform scalar point multiplication, several methods are available. However in this work, a division based addition chain has been proposed to minimize the number of additions which are involved in kP. Further, to increase the speed of the kP, fuzzy modular arithmetic is used. The application of division based addition chain and fuzzy modular arithmetic in ECC is presented in chapter 9.

The implementation methodology to run the framework is presented in chapter 10. This is illustrated by providing some sample screenshots. The generated results highlights the time taken for encryption and decryption operations of cryptographic
algorithms, the generation of producing the ciphertext from the plaintext for different file size messages, and the level of security before and after the incorporation of the proposed models in the existing cryptographic algorithms. Some samples outputs are also shown in this chapter.

The results obtained from the implementation of these models in the cryptographic algorithms are summarized. Based on the summarized result, a chart has been drawn for each model. The reason for variations in time and whether the proposed models will really increase the security level or not are also analysed in chapter 11. Finally, the thesis concludes at chapter 12 with the summary of the contribution along with the future extensions of the work.