CONSTRUCTION AND APPLICATIONS OF CRYPTOGRAPHICALLY SECURE CHAOTIC NONLINEAR COMPONENT FOR BLOCK CIPHERS

By

MAJID KHAN

Department of Mathematics
Quaid-i-Azam University, Islamabad
PAKISTAN
2015

CONSTRUCTION AND APPLICATIONS OF CRYPTOGRAPHICALLY SECURE CHAOTIC NONLINEAR COMPONENT FOR BLOCK CIPHERS

By

MAJID KHAN

Supervised

By

Prof. Dr. TARIQ SHAH

Department of Mathematics
Quaid-i-Azam University, Islamabad
PAKISTAN
2015

CONSTRUCTION AND APPLICATIONS OF CRYPTOGRAPHICALLY SECURE CHAOTIC NONLINEAR COMPONENT FOR BLOCK CIPHERS

This thesis is submitted to the Department of Mathematics,
Quaid-i-Azam University, Islamabad, in partial fulfillment of
the requirement for the degree of

Doctor of Philosophy
in
Mathematics
by
MAJID KHAN

Department of Mathematics
Quaid-i-Azam University, Islamabad
PAKISTAN
2015

Abstract

During the most recent decade, web activities have turned into an essential part of many individuals' life. As the quantity of these activities increases, so does the individual data about the clients, that is stored in electronic structure and is normally exchanged by utilizing open electronic means. This makes it practical, and frequently simple, to gather, exchange and process a vast quantity of information about a person. As a result, the requirement for instruments to secure such data is convincing. In this connection, Boolean functions and S-boxes (substitution boxes) are vital mechanisms of an information security system. These two important components are linked by function quantity. That is, an S-box is in general comprised of several distinct output Boolean functions, but if it is mapped to just a single bit, is identical to a Boolean function. Boolean functions are frequently used in the secret and public key block ciphers production as these functions are well appropriate for receiving bits of linear feedback shift registers as input in order to join them as strongly as possible to generate the single secret key stream. Furthermore, Boolean functions have also exhibited some significant properties, which are essential to oppose the classic kind of attacks, so these functions are an important component in almost all block ciphers. Cryptography, watermarking and steganography are widely used techniques for information hiding in securing communication across the internet and mobile transmission. Cryptography scrambles information so that it cannot be understood. Stenography attempts to prevent suspecting the existing of the data by an unintended recipient. Digital watermarking provides copyright protections by hiding rightful information for declaring ownership. The fundamentals of Boolean functions, cryptography, watermarking and steganography, which will be helpful for successive parts, are given in chapters 2 and 3 respectively. Along with this discussion, it will be an injustice not to pay tribute to the pioneers of computer and information sciences. Therefore, chapter 1 sheds lights on a brief account of the fathers of computer and information sciences namely George Boole and Claude Shannon which is published in **Nonlinear Engineering (DE GRUYTER, Germany)**.

The objective of information security is to obscure the information present in the original data to secure the encrypted information. The integral part of creating confusion is the introduction of randomness in data at the output. The random behavior of chaotic systems exhibits desirable properties suitable for nonlinear dynamic systems, such as, substitution process in a cipher without independent round keys. The chaotic systems are highly sensitive to initial conditions and exhibit random behavior, which is deterministic if initial information is available, and in the absence of this initial information, the system appears to be random to an observer. These properties are desirable and attractive in the design of cryptographic systems. The application of chaotic sequences to the construction of substitution boxes (S-boxes) used in Advanced Encryption Standard (AES) capable of creating confusion and applying diffusion of the original data. To build a bridge between chaos and cryptography, we have combined chaotic systems with linear functional transformation in order to produce large numbers of substitution boxes with low computational complexity and high confusion and diffusion competences. These matters of chapter 4 are published in **Nonlinear Dynamics (Springer-USA).**

We have endeavored to streamline the encryption prepare by diminishing the computational many-sided quality while in the meantime expanding the encryption quality. The proposed encryption algorithm in chapter 5 is focused around chaotic binary Boolean function in which different trajectories are all the while utilized to get a successful and quick system for securing information. The chaotic structure gives extensive key space that can be used to encode information by utilizing

chaotic maps. The system trajectory is exceptionally touchy to slight changes in the key; along these lines with the information of inexact estimations of the key, the cryptanalysis can't extricate helpful data. The main description of chapter 5 is available in **Neural Computing & Applications (Springer Verlag-London).**

The study of algebraic structures which are compatible for real time applications is always an interesting area of research for investigators. As there is always a corner for improvement in any system, therefore we have proposed a new method to design a substitution box (S-box) for the cryptographic system. The S-box substitutes the original data in the plaintext and provides the diffusion properties while maintaining high entropy levels. This process resembles the nonlinear transformation and the design of S-box must render high randomness in the encrypted data. We have used the exponential map as a thresholding function which is embedded in Galois field of modulo classes and two dimensional Tinkerbell chaotic maps for image encryption applications in chapter 6. The major components of chapter 6 are published in **Neural Computing & Applications (Springer Verlag-London).**

In the modern age, chaos-based protected communication has obtained considerable devotion since it suggests potential advantages over conventional methods due to its simplicity and high level of unpredictability. In the literature, many chaotic systems have been presented, but few have been used in cryptography. In the block cipher system, the plaintext is distributed into the blocks and the ciphering is carried out for the complete block. Two wide-ranging ideas of block ciphers which were proposed by Shanon are diffusion and confusion. Diffusion is scattering of the effect of plaintext bits to ciphertext bits with a target to obscure the statistical configuration of plaintext. Confusion is a transformation in which alterations dependency of information of ciphertext is on the information of plaintext. In most cipher structures, the diffusion and confusion are attained by means of round recurrence. Modern block encryptions comprise of four conversions: substitution, permutation, mixing and key adding. A number of famous block ciphers are of substitution-permutation (SP) category. S-boxes are used in such cipher structures as the essential nonlinear element. A robust block cipher must be hardy to numerous attacks, such as linear and differential cryptanalysis. In SP systems, this is normally reached if the S-boxes used to satisfy a number of measures. The S-box functioning in encryption procedure could be selected under the control of key, as a substitute of being static. Several random keys-dependent and bijective S-boxes are generated for encryption applications, which satisfy selected standards. In chapter 7, we have suggested new chaos-based S-boxes that are simply a combination of Hénon chaotic map and symmetry group S_8, which enhanced the confusion and diffusion capability of proposed designed block cipher. We have used our designed chaos-based S-boxes in image encryption application and investigate the texture features of second order. This detailed segment of chapter 7 has got its place in **Neural Computing & Applications (Springer Verlag-London).**

A CAPTCHA (an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a kind of test reaction test utilized as a part of figuring to figure out if or not the client is human. The term emerged in 2000 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper of Carnegie Mellon University and John Langford of IBM. The most well-known kind of CAPTCHA was initially designed by Mark D. Lillibridge, Martin Abadi, Krishna Bharat and Andrei Z. Broder. This type of CAPTCHA obliges that the client sort the letters of a twisted picture, in some cases with the expansion of a clouded arrangement of letters or digits that show up on the screen. Since the test is controlled by a machine, rather than the standard Turing test that is directed by a human, a CAPTCHA is once in a while portrayed as a reverse Turing test. This term is equivocal in light of the fact that it could likewise mean a Turing test in which the members are both endeavoring to demonstrate they are the machine. A novel construction and application of chaotic S-boxes for CAPTCHA are discussed in chapter 8 and published in **Signal image and video processing (Springer-USA).**

The term digital watermarking was initially suggested in 1992, when Tirkel exhibited two watermarking systems to conceal the watermark information in the images. The achievements of the internet are multidimensional. It is financially savvy and has mainstream advanced recording and capacity gadgets. It guarantees higher data transmission and nature of administration for both wired and remote systems have made it conceivable to make, recreate and transmit content in an easy way. However, the assurance and implementation of protected innovation rights for computerized media have turned into a critical issue. Digital watermarking is that innovation that guarantees to provide security, information verification and copyright insurance to the computerized media.

Advanced watermarking will be the installing of a sign and mystery data (i.e. Watermark) into the computerized media such as image and sound feature. Later the implanted data are discovered and concentrated out to uncover the genuine manager/personality of the computerized media. Watermarking is utilized for taking after reasons; proof of ownership (copyrights and IP assurance), copying prevention, broadcast monitoring, authentication, and data hiding. Chapter 9 dissects the key innovations of digital watermarking and investigates the application in the advanced image copyright insurance and finally got accepted in **Neural Computing and Applications (Springer-USA).**

In chapter 10, we have constructed new S-boxes which are based on class of finite rings which includes Galois rings instead of a traditional Galois field. These two structures are completely new and apply these S-boxes in image encryption and watermarking. The strength of proposed image encryption and watermarking based on finite chain rings is verified through statistical analysis.

Because of copyright infringement, forging, falsification, and misrepresentation, transmitting the computerized information in open systems such as internet which is not reliably shielded. Consequently, for securing the mystery information numerous methodologies are forward for ensuring key advanced information. Cryptographic techniques are utilized for transmitting the mystery information scrambled by cryptosystems and utilized for mystery correspondence. The pointless manifestation of the scrambled information may draw the clue to hackers. This classified information can be secured by utilizing data, concealing methods, such as watermarking and steganography, which shrouds the secret data into a cover data and create a stago object. Watermarking is utilized for screen checking, copyright guard, following exchange and comparative exercises. Conversely, steganography is utilized basically for secret correspondences. This system undetectable modifies a cover media to veil an incognito message. Subsequently, it can cover up the extreme presence of disguised correspondences. For further security, a cryptographic system is utilized before implanting procedure. Steganography is broadly classified into spatial and frequency domain techniques. The spatial domain techniques involve encoding at the LSBs level. Least Significant Bit Substitution (LSB) is the most commonly used stenographic technique. The basic concept of Least Significant Bit Substitution includes the embedding of the secret data at the bits which having minimum weighting so that it will not affect the value of the original pixel. In frequency domain, we find a way to hide information in areas of the image which is scarcely visible to compression, cropping, and image processing. In chapter 11, we combined S-boxes which are the most important object of symmetric cryptography along with the information hiding scheme namely steganography to provide protection against digital security terrorizations.

The design of public key cryptography (PKC) was presented by Diffie and Hellman in 1976, numerous PKC plans have been suggested and cracked. The trapdoor one-way capacities performed the key roles in the conception of PKC. Today, best PKC plans are focused around the apparent trouble of specific issues specifically large finite commutative rings. For instance, the trouble of solving the integer factoring problem (IFP) defined over the ring (where is the product of two large primes) structures the ground of the essential RSA cryptosystem and its variations, for example, Rabin-Williams design, LUC's strategy, Cao's concepts and elliptic curve variant of RSA like KMOV. The extended multi-measurement RSA cryptosystem, which can productively oppose low exponent assaults, is likewise characterized over the commutative ring. An alternate decent case is that ElGamal PKC family, including the fundamental ElGamal design, elliptic curve cryptosystem, discrete signature scheme (DSS) and Mccurley scheme, is focused on the difficulty of solving the discrete logarithm problem characterized over a finite field (where is a large prime), obviously a commutative ring. We have composed public key cryptosystems which are based on Abelian subgroup of general linear group over modulo classes, i.e., in chapter 12. The chief aspects of this chapter are published in **3D Research (Springer-USA).**

List of Publications

1. Majid Khan, Tariq Shah and Syeda Iram Batool, Construction of S-box based on chaotic Boolean functions and its application in image encryption , DOI: 10.1007/s00521-015-1887-y.
2. Majid Khan, A novel image encryption scheme based on multi-parameters chaotic S-boxes, Nonlinear Dynamics, 82 (2015) 527-533.
3. Majid Khan, Tariq Shah, An image encryption technique, Neural Comput & Applic, 26 (2015) 1137-1148.
4. Majid Khan, Tariq Shah, A copyright protection using watermarking scheme based on nonlinear permutation and its quality metrics, Neural Comput & Applic, 26 (2015) 845-855.
5. Majid Khan, Tariq Shah and Syeda Iram Batool, A new implementations of chaotic S-boxes in CAPTCHA, Signal, Image and Video Processing, 9 (2015) 1335-1338.
6. Majid Khan, Tariq Shah, A novel construction of substitution box with Zaslavskii chaotic map and symmetric group, Journal of Intelligent & Fuzzy Systems, 28 (2015) 1509–1517.
7. Majid Khan, Tariq Shah, A novel image encryption technique based on Hénon chaotic map and S_8 symmetric group, Neural Comput & Applic, 25 (2014) 1717-1722.
8. Majid Khan, Tariq Shah, A construction of novel chaos base nonlinear component of block cipher, Nonlinear Dynamics, 76 (2014) 377–382.
9. Majid Khan, Tariq Shah, An efficient construction of substitution box with fractional chaotic system, Signal, Image and Video Processing, 9 (2015) 1335-1338.

10. Majid Khan, Tariq Shah, Hasan Mahmood and M. A. Gondal, An efficient method for the construction of block cipher with multi-chaotic systems, Nonlinear Dynamics, 71 (2013) 493-504.

11. Majid Khan, Tariq Shah and M. A. Gondal, An efficient technique for the construction of substitution box with chaotic partial differential equation, Nonlinear Dynamics, 73 (2013) 1795-1801.

12. Majid Khan, Tariq Shah, Hasan Mahmood, M. A. Gondal and I. Hussain, A novel technique for constructions of S-Boxes based on chaotic Lorenz systems, Nonlinear Dynamics, 70 (2012) 2303–2311.

13. Majid Khan, Tariq Shah and Ali Shahab, A bit history about the fathers of computer and information sciences, Nonlinear Engineering – Modeling and Application, 4 (2015) 39–41.

14. Majid Khan, Tariq Shah, A novel cryptosystems based on general linear group, 3D Research, 6 (2015) 1-8.

15. Majid Khan, Tariq Shah, A literature review on image encryption, 3D Research, 5 (2014) 1-29.

16. Majid Khan, Tariq Shah and Syeda Iram Batool, Texture analysis of chaotic coupled map lattices based image encryption algorithm, 3D Research, 5 (2014) 1-5.

17. Majid Khan, Tariq Shah, A Novel Statistical Analysis of Chaotic S-box in Image Encryption, 3D Research, 5 (2014) 1-16.

18. T. Shah, I. Hussain, M. A. Gondal, W.A. Khan and Majid Khan, Construction of new S-box using a linear fractional transformation, World applied Sciences Journal, 14 (2012) 1779-1785.

Contents

Chapter 1

A Brief Description about the Fathers of Computer and Information Sciences

In this chapter, we are mainly presenting a tribute to the fathers of computer and information sciences, George Boole and Claude Elwood Shannon with their hardships and achievements. This piece of writing also elaborates the applications of George Boole's and Claude Shannon's works in different disciplines.

1.1 The Contributions of George Boole

The most reliable biography of Boole is "George Boole: His Life and Work", by Desmond Machale (Boole Press, 1985) [1]. We will utilize both this book [1] and the life story composed by O'connor and Robertson for the Mactutor History of Mathematics document [2]. Also, we have taken some of historical memories from the book of Thomas [3].

George Boole was born in November 1815 in Lincoln, England. His father was an ordinary tradesman. He gave Boole his first mathematics lessons and planted in him the passion of learning. A family friend ,who was a local bookseller, helped him learn basic Latin. By the age of 12, Boole was beginning to translate Latin poetry. By 14, the adolescent Boole was fluent in French, German, and Italian as well. His love for poetry and novels was remarkable. His capabilities in higher mathematics did not indicate until he was 17 years of age . He read his maiden progressed arithmetic book, in particular Lacroix's Differential and Integral Calculus. Since his father's business fizzled, he was compelled to earn his bread to look after his family. At 16, he turned into an assistant master in a non-public school at Doncaster, and before reaching 20 years of age, he opened his own school. In 1838, Boole was offered to assume control over the Hall's Academy in Waddington, after its organizer, Robert Hall, passed on.

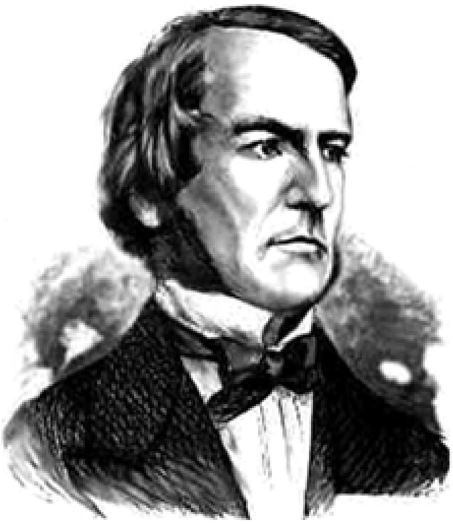


Fig. 1.1: George Boole [16].

The members of his family also shifted to Waddington and assisted him to run the school. Utilizing mathematical journals obtained from the nearby Mechanic's Institute, Boole perused over the Principia of Isaac Newton and the works of French mathematicians Pierre-Simon Laplace (1749–1827) and Joseph Louis Lagrange (1736–1813). In the wake of realizing what these creators formerly composed, Boole, at 24, distributed his first paper (Studies on the notion of analytical transformations) in the Cambridge Mathematical Journal (CMJ). It started a kinship between George Boole and the editor of CMJ, Duncan F. Gregory, which sustained until the unexpected demise of Gregory in 1844. Gregory motivated Boole to study algebra. In view of his family's financial circumstance, Boole was not able to act upon Gregory's advice to attend courses at Cambridge. Truly, in the late spring of 1840, he opened a residential school in Lincoln and again the entire family shifted with him.

When his father kicked the bucket, Boole assumed the charge of Mathematics Professorship at Queen's College in 1849 , where he stayed and educated for whatever is left of his life. It was there he saw Mary Everest, a niece of Sir George Everest. She was 17 years more youthful than him, however, they turned into companions quickly. George started giving Mary lessons on the differential calculus, and in 1855, after her father passed away, Mary wedded George Boole. They were very upbeat together and five girls were conceived: Mary Ellen (b. 1856), Margaret (b. 1858), Alicia (later Alicia Stott) (b. 1860), Lucy Everest (b. 1862), and Ethel Lilian (b. 1864).

There are 50 articles that contain the work of Boole , besides a couple of different publications. A rundown of Boole's diaries and papers, on logical and mathematical topics, is found in the Catalog of Scientific Memoirs distributed by the Royal Society, and in a volume on differential equations (altered by I. Todhunter). Boole composed 22 articles in the Cambridge Mathematical Journal and its successor, the Cambridge and Dublin Mathematical Journal. He further wrote 16 papers in the Philosophical Magazine, six diaries in the Philosophical Transactions (The Royal Society), and a couple of others in the Transactions of the Royal Society of Edinburgh and of the Royal Irish Academy, in the Bulletin de l' Academie de St-Petersbourg (in 1862, under the alias. Boldt), and in Crelle's Journal, and a paper on the mathematical basis of logic published in the Mechanics Magazine (1848). In 1844, the Royal Society bestowed him with a decoration for his commitments to analysis, as a result of his work on utilizing algebra and calculus to analyze infinitely small and large figures. Analytics of thinking, which Boole was engrossed with, thought that it was' route into his 1847 work, The Mathematical Analysis of Logic, that developed the work of the German mathematician Gottfried Wilhelm Leibniz (1646–1716) and pushed the thought that logic was a mathematical discipline, instead of philosophy. This paper won him the profound respect of the recognized logician Augustus de Morgan, and a spot among the talent of Ireland's Queen's.

The publication of Boole's "An Investigation into the Laws of Thought", on which the Mathematical Theories of Logic and Probabilities are based, is considered to be his most critical work in 1854. Boole advanced logic in an alternative way, lessening it to a straightforward algebra, fusing rationale into science, and establishing the frameworks of the now acclaimed parallel methodology. Logical statements are presently spoken to utilizing a scientific structure called as a part of his honor Boolean Algebra.

Boole's virtuoso was immensely perceived and he got honorary degrees from the universities of Oxford and Dublin. He was chosen a Fellow of the Royal Society in 1857. For his work in the long run guided individuals to step on the Moon, it is just a compliment that Boole is the name of a lunar cavity.

It was an ominous day in 1864, Boole was strolling from his home to the college and was getting in a downpour storm. Yet he taught in wet wears and caught a cold. It was a black day for mathematics when he passed away, he was just 49 years of age.

1.2 The Role of Claude Elwood Shannon to Revive Boole's Works

Two great accounts of Shannon were composed by Sloane and Wyner, and by Liversidge in the altered book by Sloane and Wyner containing Shannon's gathered papers [6]. Boole's work on mathematical rationale was censured and/or disregarded by his counterparts, aside from an American rationalist, Charles Sanders Peirce (1839–1914), who gave a discourse at the American Academy of Arts and Sciences, portraying Boole's thoughts. Peirce put in more than 20 years chipping away at these thoughts and their applications in electronic hardware; at last, he composed a hypothetical electrical logic circuit. Sadly, Boolean algebra and Peirce's work remained generally obscure and unused for decades, until the 1940s, when a youth called as Claude Elwood Shannon found Boole's and Peirce's works and perceived their importance to hardware design.

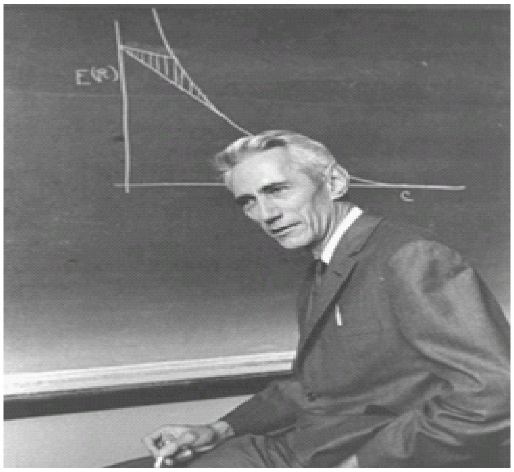


Fig. 1.2: Claude Elwood Shannon [16].

Claude E. Shannon opened his eyes in Petoskey, Michigan, on April 30, 1916. His father was a representative and, for a period, Judge of Probate. His mother was a dialect instructor and for various years Principal of Gaylord High School, in Gaylord, Michigan. Shannon stayed with Gaylord until he was 16 when he moved on from secondary school. He demonstrated a liking for science and mathematics and kept himself occupied by building model planes, a radio-controlled model watercraft, and a broadcast framework for a companion's home a large portion of a mile away [6].

Taking after his sister, in 1932, he took admission in University of Michigan (UM), where he was exposed to the work of George Boole. Shannon moved from UM in 1936 with double Bachelor's Degrees of Science in Electrical Engineering and Science in Mathematics. Immediately, tolerating an examination associate position at Massachusetts Institute of Technology to help himself, he started his graduate studies. In 1940, he completed his master degree in Electrical Engineering and a Phd in Mathematics. His Master's thesis "A Symbolic Analysis of Relay and Switching Circuits" is a venture to utilize Boole's algebra to investigate transfer exchanging circuits, while his doctoral thesis manages populace hereditary qualities. A rendition of his Master's proposal was distributed in Transactions of the American Institute of Electrical Engineers in 1940, and earned him the Alfred Noble (American Institute of Engineers) Award.

In the wake of putting in a year at the Institute for Advanced Study, in 1941 Shannon joined AT& Bell Telephones in New Jersey as an exploration mathematician to take a shot at fire-control frameworks and cryptography. He stayed affiliated with Bell Laboratories until 1972, yet took up different positions (MIT; Center for the Study of the Behavioral Sciences in Palo Alto; Institute for Advanced Study in Princeton, Visiting Fellow at All Souls College, Oxford; University of California; the IEEE; and the Royal Society).

In 1949 Shannon entered into a wedlock with Mary Elizabeth Moore. They were blessed with three boys: Robert, James, Andrew Moore; and one little girl Margarita.

In one of his most critical works, A Mathematical Theory of Communication [4], Shannon established the subject of data, hypothesis and he proposed a straight schematic model of an interchange framework. This was a progressive thought as there was no more any requirement for electromagnetic waves to be sent down a wire. One could convey rather, by sending groupings of 0 and 1 bits. In the following year, he composed an alternate key paper, Communication Theory of Secrecy Systems [5], which is the first investigation of cryptography. It was focused around classified chip away at mystery frameworks embraced by Shannon in the final year of World War II. Shannon kicked the bucket in 2001 after a long battle with Alzheimer's disease.

1.3 Conclusion

In this chapter, we have discussed two well-known scientists Boole and Shannon. Their contributions still give new inspiration to computer science and modern information security which is a growing area of research nowadays. Such geniuses could never be seen even after hundreds of years. Their work directly influenced channel coding, information security, cryptography and cryptanalysis. This chapter is written only to acknowledge these two highly intellect personalities of computer and information sciences.

Chapter 2

Boolean Functions and Cryptographic Properties of Substitution Boxes

Nowadays, society is tightly surrounded by the sphere of the information era, which is classified by scholar assets and is utilizable inside data being considered exceptionally precious. Informative data exists and is used in various forms as economic, official (documents), martial, and political. The safety of this data during transfer and saving, and in routine practice is very important because its compromise might affect in the disclosure of marketing, financial loss, or armed forces top secrets, and even the loss of life. Cryptology is a significant way used in the Survey of data protection. Three most important types of security are presented by cryptology through the utilization of appropriate and healthy structured cryptosystems.

These systems are known as confidentiality, integrity, and authentication. Confidentiality is offered by guaranteeing that secret data is kept personal from unofficial disclosure. Integrity is offered by making sure the secret data has not been altered, even coincidentally, during production or storage. Authentication is the procedure of assessment that the dispatcher of the data is properly recognized and legitimate.

The encryption schemes are often classified by some aspects such as the philosophy of their key distribution and the dimension of their input stream. Symmetric encryption algorithms have a common secret key allocated to senders and receivers, while asymmetric cryptosystems use dissimilar keys for enciphering and deciphering. Symmetric cryptosystems have two main branches, block or stream ciphers, where the input data to the cryptosystem catches the shape of either blocks or unbroken bit streams, respectively. Another kind of encryption scheme is a hash function, which squeezes information in a digest form for the sake of integrity or authentication.

Cryptosystems are key marks for an attacker desiring to compromise the secret data being guarded by a security algorithm. In procession with the three types of security requirements cited above, the usual motivations of an attacker is to disclose secret information, to illegally and underhandedly alter information, and to falsely adopt an identity. Furthermore, an attacker possibly will try to eliminate evidence, or even add fake evidence that a result or transaction has taken place.

Compromising a cryptosystem which is responsible to protect the secret information can either directly permits these events to happen, or ultimately weaken a different ingredient of the scheme to allow these actions to later take place. Prevailing accessible cryptanalytic attacks in opposition to cryptosystems have conformed to be unbeaten under these conditions.

The general strength of an encryption algorithm is reliant on the strength of the individual components, such as the authentication scheme, the secret key management scheme, the information saving scheme, the cryptosystem, the policies and methods, etc. Likewise, the whole strength of a security system is reliant on the potency of its individual mechanism. A flaw in any of the individual processes may lead to a shattering breakdown in the entire security system.

S-boxes (substitution boxes) and Boolean functions are vital mechanisms of cryptosystems. These two important components are linked by function quantity. That is, an S-box is in general comprised of several distinct output Boolean functions, but if it is mapped to just a single bit, is identical to a Boolean function.

Boolean functions are frequently used in the secret key stream production procedure of stream ciphers as these functions are well appropriate for receiving bits of linear feedback shift registers as input in order to join them as strongly as possible to generate the single secret key stream. Furthermore, Boolean functions have also exhibited some significant properties, which are essential to oppose the classic kind of attacks, so these functions are an important component in almost all stream ciphers.

The substitution box (S-box) is the key component used in many block ciphers. It offers a way of substituting various blocks of bits for a totally dissimilar set of output bits. One thing which is very important is the use of secure substitution boxes (those which hold excellent encryption properties) so the substitution indicates a confused association between input and output bits of the substitution box. One of the main functions of the substitution box, when used in iterative round function, is to enhance the effort required to explore any statistical structure in the secure data.

Substitution boxes are capable to provide the safety of an encryption algorithm by possessing excellent encryption properties. Constructing secure S-boxes to use them in different cryptosystems for the sake of increasing their security is a current research problem. This is mainly so because the cryptanalytic system turns out to be more refined, and with the improvement of computer technology that contributes equally supporting and against secure communication.

The strength of substitution boxes has a major bearing on secure communication. However, bigger functions usually need additional computational time and effort in order to explore their flaws, so we gain a good computational complexity enhancement when trying to find large functions with remarkably excellent measures of attractive encryption properties. This includes an additional part of complexity to the research problem.

2.1 Review of Substitution Box Theory

The research study reported in this dissertation needs understanding about Boolean function and previously constructed S-box theory. Such literature review is important not simply for connecting the theoretical ideas to handy applications, but as well to recognize the importance of the research and where this study is connected to the field of secure communication. Therefore, fundamental background is presented in this paper by stating a number of important long established and well known definitions, theorems and formulae.

First we presented some important concepts about Boolean function. This includes discussions about cryptographic properties of Boolean functions, as well as the nature of the association among different cryptographic properties. After that we present theory about S-boxes, including the definition of various cryptographic properties of S-boxes. At the end, a concise review of some well known S-boxes is presented.

2.1.1 Boolean Function Theory

The study of Boolean algebra is a widespread and generalized area in itself. This section presents a small literature survey of Boolean function theory. To a certain extent, the survey provided in this section is a complete organization of that which is required for the reader to completely be aware of the research presented in this dissertation. Particularly, we have discussed some important cryptographic properties which are applicable to this work.

2.1.2 Properties of Boolean Functions

The purpose of this section is to make some preliminary definitions on Boolean functions. Let $GF(2)^n$ be the vector space of dimension n over the two-element Galois field $GF(2)$. $GF(2)^n$ consist of 2^n vectors written in a binary sequence of length n. The vector space is equipped with the scalar product $< ., . >: GF(2)^n \times GF(2)^n \to GF(2)$

$$< u, v >= \oplus_{j=1}^{m} u_j.v_j, \tag{2.1}$$

where the multiplication and addition $\oplus$ are over $GF(2)$. However, if additions are performed in the real numbers, then it is clear from the context.

Definition 1 *A Boolean function of n variables is a function $g : GF(2)^n \rightarrow GF(2)^n$ (or simply a function on $GF(2)^n$). The $(0,1)$-sequence is defined by $(g(\alpha_0), g(\alpha_1), ..., g(\alpha_{2^n-1}))$, also called the truth table of g, where $\alpha_0 = (0,0,...,0), \alpha_1 = (0,0,...,1), ..., \alpha_{2^n-1} = (1,1,...,1)$, ordered by lexicographical order.*

Definition 2 *A vector Boolean function is a function that maps a Boolean vector to another Boolean vector:*

$$\zeta : GF(2)^n \rightarrow GF(2)^m. \tag{2.2}$$

This vector Boolean function has n input bits and m output bits. A vector Boolean function can be specified by its definition table: an array containing the output value for each of the 2^n possible input values. Each bit of the output of a vector Boolean function is itself a Boolean function of the input vector. These are the coordinate Boolean functions of the vector Boolean function.

Definition 3 *A vector Boolean transformation is a vector Boolean function with the identical number of input bits as output bits.*

Definition 4 *A vector Boolean permutation is an invertible vector Boolean transformation and maps all input values to different output values. There are 2^{m2^n}, n bit to m bit vector Boolean functions. A random n bit to m bit vector Boolean function is a function selected at random from the set of 2^{m2^n} different n bit to m bit vector Boolean functions, where each function has the same probability of being chosen. A random vector Boolean function can be obtained by pulling its definition table with 2^n random m bit values.*

Definition 5 *The logical negation or complement of a Boolean function g is defined by $\overline{g} = g \oplus 1$.*

Definition 6 *A linear Boolean function is denoted by*

$$L_\alpha(x) = \alpha_1 x_1 \oplus \alpha_2 x_2 \oplus ... \oplus \alpha_n x_n, \tag{2.3}$$

where $\alpha_i x_i$ denotes the bitwise AND of the $i-th$ bits of α, x and $\oplus$ denotes bitwise XOR.

Definition 7 *The set of affine Boolean functions is the set of linear Boolean functions and their complements*

$$A_{\alpha,c} = L_\alpha(x) \oplus c, \tag{2.4}$$

where $x \in GF(2)^n$. The sequence of an affine (or linear) function is called an affine (or linear) sequence.

Definition 8 *The set of all single valued Boolean functions is denoted by*

$$G_n = \{g \,|g : GF(2)^n \rightarrow GF(2)\}. \tag{2.5}$$

The subset of all affine Boolean functions in the space G_n is denoted by

$$A_n = \{\beta \,|\beta : \textit{is affine and } \beta \in G_n\}. \tag{2.6}$$

We define the subset of all linear Boolean functions in the space $GF(2)^n$ by

$$L_n = \{\alpha|\alpha : \textit{is linear and } \alpha \in G_n\}. \tag{2.7}$$

Remark 9 *The set of all affine functions consist of the linear functions and their negations.*

Remark 10 *The cardinalities of the above sets are easily observed as*

$$|G_n| = 2^n,\ |A_n| = 2^{n+1},\ |L_n| = 2^n. \tag{2.8}$$

Definition 11 *To each Boolean function $g : GF(2)^n \to GF(2)$,we associate its sign function, or character form, denoted by $\widehat{g} : GF(2)^n \to \mathbb{R}^* \subseteq \mathbb{C}^*$,and defined by*

$$\widehat{g}(x) = (-1)^{g(x)}. \tag{2.9}$$

The $(1, -1)-$sequence is defined by $((-1)^{g(\alpha_0)}, (-1)^{g(\alpha_1)}, ..., (-1)^{g(\alpha_{2^n-1})})$, where α_j are defined in definition 1.

Proposition 12 *[54, Proposition 2.6]If g and h are Boolean functions on $GF(2)^n$, the following statements holds:*

1. $\widehat{g \oplus h} = \widehat{g}\widehat{h}$,
2. $\widehat{gh} = 1 + \widehat{g} + \widehat{h} - \widehat{g}\widehat{h}$.

Proof. 1. By definition, we have

$$\widehat{g \oplus h} = (-1)^{g \oplus h} = (-1)^g(-1)^h = \widehat{g}\widehat{h}. \tag{2.10}$$

2. This claim can be prove with the help of the following observation , $\widehat{g} = 1 - 2g$ i.e.,

$$\begin{aligned} 1 + \widehat{g} + \widehat{h} - \widehat{g}\widehat{h} &= 1 + (1 - 2g) + (1 - 2h) + (1 - 2g)(1 - 2h), \\ &= 2 - 4gh, \\ &= 2\,(1 - 2gh)\,, \\ &= 2\widehat{gh}. \end{aligned} \tag{2.11}$$

■

Definition 13 *The Hamming-weight of a Boolean function $g : GF(2)^n \rightarrow GF(2)$, is the number of 1's in the truth table of g. Next, we introduce the notion of distance between two Boolean functions.*

Definition 14 *For two Boolean functions $g, h : GF(2)^n \rightarrow GF(2)$, we define the Hamming-distance as the number of arguments where g and h differ, that is*

$$d(g, h) = \#\{x \in GF(2)^n \mid g(x) \neq h(x)\}. \tag{2.12}$$

In other words, the Hamming-distance is the number of $1's$ in the truth table of $g \oplus h$. We can also express the Hamming-distance in terms of the Hamming-weight $d(g, h) = \mathbf{wt}(g \oplus h)$ as It is simple to show that the Hamming-distance d is the metric on $GF(2)^n$. It follows by notion that $d(g, h)$ equals to the numbers of the entries that are needed to turn g to h. Thus $d(g, h)$ is zero if and only if $g = h$.

Definition 15 *The support of a Boolean function g is defined as*

$$\mathbf{supp}(g) = \{x \in GF(2)^n \mid g(x) = 1\}\ . \tag{2.13}$$

The Hamming-weight can also be expressed in the notions of the Hamming-distance and the support of a Boolean function as:

$$\mathbf{wt}(g)\mathbf{=}d(g, 0)\mathbf{=}\ \mathbf{supp}(g)\mathbf{.} \tag{2.14}$$

Definition 16 *A $(0, 1)-$sequence ($(1, -1)-$sequence) is called balanced if it contains an equal number of zeros and one (ones and minus ones). A function is balanced if its sequence is balanced that is* $\mathbf{wt}(g) = 2^{n-1}$.

Definition 17 *The imbalance $\mathbf{Imb}(g)$ of a Boolean function g is the number of inputs that maps to* 0 *minus the number of inputs that maps to* 1 *divided by two. The imbalance can have any integer value and ranges from -2^n to 2^n. We have*

$$\mathbf{Imb}(g) = 1/2(\#\{a\,|g(a) = 0\} - \#\{a\,|g(a) = 1\}). \tag{2.15}$$

A Boolean function with imbalance 0 is called balanced.

Definition 18 *Two Boolean functions g, h on $GF(2)^n$ are called (affinely) equivalent if $g(x) = h(Ax \oplus b)$ where $a, b \in GF(2)^n$ and A is $n \times n$ is nonsingular matrix. If no such transformation exists, then g and h are called inequivalent.*

Definition 19 *The autocorrelation function $\widehat{r_{\widehat{g}}}(a)$ with a shift $a \in GF(2)^n$ is defined as*

$$\widehat{r_{\widehat{g}}}(a) = \sum_{x \in GF(2)^n} \widehat{g}(x).\widehat{g}(x \oplus a). \tag{2.16}$$

Definition 20 *Let g be a function defined on $GF(2)^n$. Let $a \in GF(2)^n$ is called a linear structure of $a \in GF(2)^n$ if*

$$\widehat{r}_{\widehat{g}}(a) = 2^n, \tag{2.17}$$

i.e., if $\widehat{g}(x).\widehat{g}(x \oplus a)$ is constant.

The set of all linear structures of a function g form a linear subspace of $GF(2)^n$. The dimension gives a measure of linearity. This measure is upper bounded by 2^n. The bound is attainable by the all zero vector in $GF(2)^n$ and follows from lemma 31. A nonzero linear structure is cryptographically undesirable.

Definition 21 *The correlation value between two Boolean functions g and h is defined by*

$$\begin{aligned} C(g,h) &= 2\Pr(g(x) = h(x)) - 1, \\ &= 2\left[\frac{2^n - d(g,h)}{2^n}\right] - 1, \\ &= \frac{2^{n+1} - 2d(g,h)}{2^n} - 1, \\ &= 1 - \frac{2d(g,h)}{2^{n-1}}. \end{aligned} \tag{2.18}$$

Correlation is a rational number in the range $[-1, 1]$. From the definition, we see that the upper bound of 1 is achieved when the Hamming distance between two functions is zero. Similarly, the lower bound -1 is achieved when the Hamming distance between two functions is equal to 2^n. Correlation is an important tool in the analysis of pairs of functions, particularly in relation to the concept of imbalance in a Boolean function.

Definition 22 *The number of variables in highest order monomial with zero coefficients is called the algebraic degree.*

Definition 23 *The algebraic normal form (ANF) is an n-variables Boolean function which can be written as follows:*

$$g(x) = b_0 \oplus b_0 x_0 \oplus ... \oplus b_{01} x_0 x_1 \oplus b_{012...n-1} x_0 x_1 x_2 ... x_{n-1}, \tag{2.19}$$

where the coefficients $b \in GF(2)^n$ form the elements of the truth table of the ANF of $g(x)$. Note that each product term in the ANF is calculated by the multiplication of each of the components of that term.

Definition 24 *A Boolean function is said to be homogeneous if its algebraic normal form only contains terms of the same degree.*

Definition 25 *An n variable Boolean function $g(x)$, which contains all n variables in its ANF is called a nondegenerate function. Conversely, if $g(x)$ does not contain every variable in its ANF representation then the function is degenerate.*

Definition 26 *The algebraic degree of a Boolean function is a good indicator of the function's algebraic complexity. The higher the degree of a function, the greater is its algebraic complexity.*

Definition 27 *The algebraic degree of a Boolean function $g(x)$, denoted by $deg(g)$, is defined to be the number of variables in the largest product term of the function's ANF having a non-zero coefficient.*

Remark 28 *The algebraic normal form is not the only the representation to express a Boolean function. Also the disjunction normal form (DNF) is a possibility. Carlet and Guillot introduced yet another representation, the so called numerical normal form (NNF).*

2.2 Nonlinearity of Boolean Function

Nonlinearity is one of the most important cryptographic properties. As before, we denote with the set of all affine functions and the Hamming-distance is the number of arguments where the Boolean functions g and h differ. In addition, Pieprzyk and Finkelstein [164] introduced the notion of nonlinearity as follows:

Definition 29 *The nonlinearity of a Boolean function is denoted by and is defined as follows*

$$N_g = d(g, A_n) = \min_{\alpha \in A_n} d(g, \alpha). \tag{2.20}$$

It is obvious that the nonlinearity of an affine function is zero. If the Boolean function g is not affine, then we have $N_g > 0$ by definition. High nonlinearity is essential designing a good cryptosystem. It measures the ability of a cryptographic system using the functions to resist against being expressed as a linear set of equations and it assures resistance against linear cryptanalysis introduced by [29].

2.3 The Walsh Transform

In this section, we introduce one of the most important tools in cryptography. Namely, the Walsh transforms which is the characteristic 2 case of the discrete Fourier transform. As we shall see, the use of the Walsh transform makes the computation of nonlinearity and the other properties an easy task. Let us recall that we have the space G_n of all two-valued functions on $GF(2)^n$. The domain of G_n is an abelian group and its range elements 0 and 1 can be added and multiplied as complex numbers. Now we analyze G_n by using tools from harmonic analysis, cf. Lechner [22]. This means that we are able to construct an orthogonal basis of Fourier transform kernel functions, or also known as group characters, on G_n. The kernel functions are defined in terms of a group homomorphism $GF(2)^n$ from to the direct product of n copies of the multiplicative subgroup $\{-1, 1\}$ on the unit circle of the complex plane. We define the Walsh transform of a Boolean function as follows [54]:

Definition 30 *[54] The Walsh transform of a function g on $GF(2)^n$ is a map $\Omega : GF(2)^n \to \mathbb{R}$ defined by*

$$\Omega(g)(u) = \sum_{x \in GF(2)^n} g(x)(-1)^{<u,x>}, \tag{2.21}$$

where $< u, x >$ is the canonical scalar product. The Walsh spectrum of g is the list of 2^n Walsh coefficients given by Eq. (2.21) *as varied.*

Lemma 31 *[54, Lemma 3.2]If $u \in GF(2)^n$, we have*

$$\sum_{x \in GF(2)^n} (-1)^{<u,x>} = \begin{cases} 2^n, & if\ u = 0 \\ 0, & else. \end{cases} \tag{2.22}$$

Proof. *If $u = 0$, then all exponents are zero and therefore all summands are equal to 1. Therefore, we have 2^n summands. Now we assume that $u \neq 0$ and consider the hyperplanes $H = \{x \in GF(2)^n | < u, x >= 0\}$ and $\overline{H} = \{x \in GF(2)^n | < u, x >= 1\}$. It is obvious that these hyperplanes generates a partition of $GF(2)^n$. Furthermore, for any $u \in H$, the summand is equal to one and for any $u \in \overline{H}$, the summand is equal to -1. In addition, the number of elements in H and $\overline{H}$ are same that is 2^{n-1}. Therefore, the sum equals zero and the given statement follows immediately.* ■

Theorem 32 *[54, Theorem 3.3] The Walsh transform $\Omega : GF(2)^n \to \mathbb{R}$ is bijective and the inversion is given by:*

$$\Omega^{-1} = \Omega/2^n. \tag{2.23}$$

Hence g, can be recovered by the inverse *Walsh transform* given by

$$g(x) = \sum_{u \in GF(2)^n} \Omega(g)(u).(-1)^{<u,x>}. \tag{2.24}$$

At that point we do a short insertion about Hadamard matrices and Sylvester-Hadamard matrices. This leads us to express the Walsh transform in term of Sylvester-Hadamard matrices.

Definition 33 *The Sylvester-Hadamard matrix (or Walsh-Hadamard matrix) of order 2^n, denoted by H_n is generated by the recursive relation*

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} = H_1 \otimes H_{n-1}, \tag{2.25}$$

for $n = 1, 2, ...and\ H_0 = (1)$.

With this definition, we are able to express the Walsh transform in terms of Sylvester-Hadamard matrices, giving us $\Omega(g) = g.H_n$, since $(-1)^{<u,x>}$ is the entry on the position $(u, v) \in GF(2)^n$ in the matrix H_n. Additionally, we can easily express the inverse Walsh transform as $g = 2^{-n}\ \Omega(g).H_n$.

Next we collect some properties of the Walsh transform. The following Lemma shows the connection between the Walsh transform of two Boolean functions where one function is obtained by an affine transformation of the input coordinates.

Lemma 34 *[3] If the Boolean function can be obtained from h by an affine transformation of the input that is $h = g(Av \oplus b)$, with A an invertible matrix and $b \in GF(2)^n$, then the Walsh transform of g and h are related by*

$$\Omega(h)(u) = \pm\Omega(g)(uA^{-1}). \tag{2.26}$$

Proof. *First, we have to use the following definition*

$$\Omega(h)(u) = \sum_{v \in GF(2)^n} \Omega(h)(v).(-1)^{<u,v>} = \sum_{v \in GF(2)^n} (-1)^{<u,v>} g(Av \oplus b). \tag{2.27}$$

By setting v=$A^{-1}w \oplus A^{-1}b$ and $u' = uA^{-1}$, we get

$$\begin{aligned}\Omega(h)(u) &= \sum_{w \in GF(2)^n} (-1)^{<u,A^{-1}w>}(-1)^{<u,A^{-1}b>} g(w),\\ &= \pm \sum_{w \in GF(2)^n} (-1)^{<u',w>} g(w),\\ &= \pm\Omega(g)(u'). \end{aligned} \tag{2.28}$$

■

Furthermore, we observe the relationship between the Walsh transform of a Boolean function and its sign function which was introduced by Forre [20].

Lemma 35 *Let $\widehat{g}(x) = (-1)^{g(x)}$, then $\Omega(\widehat{g})(u) = -2\Omega(g)(u) + 2^n\delta(u)$, which is equivalent to $\Omega(g)(u) = 2^{n-1}\delta(u) - \frac{1}{2}\Omega(g)(u)$, where*

$$\delta(u) = \begin{cases} 1, & for\ u = 0, \\ 0, & else. \end{cases} \tag{2.29}$$

is the Dirac symbol.

Proof. We start from the left-hand side of the first equation and obtained

$$\begin{aligned}
\Omega(\widehat{g})(u) &= \sum_{x\in GF(2)^n} (-1)^{g(x)\oplus<u,x>}, \\
&= \sum_{x\in GF(2)^n} (1-2g(x)).(-1)^{<u,x>}, \\
&= \sum_{x\in GF(2)^n} (-1)^{<u,x>} - 2\sum_{x\in GF(2)^n} g(x).(-1)^{<u,x>}, \\
&= 2^n\delta(u) - 2\Omega(g)(u), \qquad (2.30)
\end{aligned}$$

by definition 30 and lemma 31. ■

The following lemmas provide us with some properties satisfied by the Walsh transform.

Lemma 36 *[54, Lemma 3.9]The following statements are true:*

1. $\Omega(\widehat{g\oplus 1})(u) = -\Omega(\widehat{g})(u)$,
2. *If* $h(x) = g(x)\oplus\alpha_a(x)$, *where* $\alpha_a(x) = \sum_{i=1}^{n} a_i x_i =< a, x >$ *is linear function, then* $\Omega(h)(u) = \Omega(\widehat{g})(u\oplus a)$.
3. *If* $h(x)$=$\alpha_a(x)\oplus c$ *is the affine function then* $\Omega(\widehat{g\oplus h})(u) = (-1)^c\Omega(\widehat{g})(u\oplus a)$.

Proof.

$$\begin{aligned}
1. \qquad \Omega(\widehat{g\oplus 1})(u) &= \sum_{x\in GF(2)^n} (-1)^{g(x)\oplus 1\oplus<u,y>}, \\
&= -\sum_{x\in GF(2)^n} (-1)^{g(x)\oplus<u,y>}, \\
&= -\Omega(\widehat{g})(u). \qquad (2.31) \\
2. \qquad \Omega(h)(u) &= \sum_{x\in GF(2)^n} (-1)^{g(x)\oplus\alpha_a(u)\oplus<u,x>}, \\
&= \sum_{x\in GF(2)^n} (-1)^{g(x)\oplus<<u\oplus a>,x>}, \\
&= \Omega(\widehat{g})(u\oplus a). \qquad (2.32) \\
3. \qquad \Omega(\widehat{g\oplus h})(u) &= \sum_{x\in GF(2)^n} (-1)^{g(x)\oplus\alpha_a(u)\oplus<u,x>}, \\
&= (-1)^c \sum_{x\in GF(2)^n} (-1)^{g(x)\oplus<<u\oplus a>,x>}, \\
&= (-1)^c\Omega(\widehat{g})(u\oplus a). \qquad (2.33)
\end{aligned}$$

■

The addition of an affine function causes, except for the sign, a permutation of the spectrum.

Corollary 37 *[54, Corollary 3.10]In particular* $\Omega(\widehat{g})(u)$ *is always even and we have*

$$-2^n \leq \Omega(\widehat{g})(u) \leq 2^n. \qquad (2.34)$$

A classic property of the Walsh transform is to be an isomorphism from the set of the sign functions on $GF(2)^n$, endowed with the so-called convolution product (denoted by $*$), into this same set, endowed with the usual product. The notion of the convolution is given within the next definition.

Definition 38 *Let g and h be any Boolean functions on $GF(2)^n$. The Convolution of g and h is defined by*

$$(g * h)(x) = \sum_{y \in GF(2)^n} g(y)h(x \oplus y). \tag{2.35}$$

Proposition 39 *[54, Proposition 3.12]Let g and h be any Boolean function on $GF(2)^n$. We have*

$$\Omega(g * h) = \Omega(g) * \Omega(h). \tag{2.36}$$

Consequently

$$\Omega(g) * \Omega(h) = 2^n \Omega(g.h). \tag{2.37}$$

Proof. We have

$$\begin{aligned}
\Omega(g * h) &= \sum_{x \in GF(2)^n} (g * h)(x).(-1)^{<u,x>}, \\
&= \sum_{x \in GF(2)^n} \sum_{y \in GF(2)^n} g(y)h(x \oplus y).(-1)^{<u,x>}, \\
&= \sum_{x \in GF(2)^n} \sum_{y \in GF(2)^n} g(y)h(x \oplus y).(-1)^{<u,x> \oplus <u,x \oplus y>}, \\
&= \left(\sum_{y \in GF(2)^n} g(y)(-1)^{<u,y>} \right) \left(\sum_{x \in GF(2)^n} h(x \oplus y)(-1)^{<u,x \oplus y>} \right), \\
&= \left(\sum_{y \in GF(2)^n} g(y)(-1)^{<u,y>} \right) \left(\sum_{x \in GF(2)^n} h(x)(-1)^{<u,x>} \right), \\
&= \Omega(g).\Omega(h).
\end{aligned} \tag{2.38}$$

■

Thereby, the first equality is proven. We recall the property $\Omega(\Omega(g)) = 2^n g$. Therefore, we obtain $\Omega(\Omega(g) * \Omega(h)) = 2^{2n} g.h$. Again, using the property, we get $\Omega(g) * \Omega(h) = 2^n \Omega(g.h)$. Using Eq. (2.36) applied at $x = 0$ gives

$$\Omega(g) * \Omega(h)(0) = 2^n \Omega(g.h)(0) = 2^n \sum_{x \in GF(2)^n} g(x)h(x) = 2^n g * h(0). \tag{2.39}$$

Taking $g = h$ in Eq. (2.38) , we obtain Parseval's equation. Parseval's equation will be a useful tool to prove some of following results.

Corollary 40 *(Parseval's equation) For any Boolean function g in n variables, the following equations holds*

$$\sum_{u \in GF(2)^n} \left(\Omega(\widehat{g})\right)(u)^2 = 2^{2n}. \tag{2.40}$$

Proof.

$$\begin{aligned}
\sum_{u \in GF(2)^n} \left(\Omega(\widehat{g})\right)(u)^2 &= \sum_{u \in GF(2)^n} \sum_{x \in GF(2)^n} \sum_{y \in GF(2)^n} (-1)^{g(x) \oplus g(y) < u, x \oplus y >}, \\
&= \sum_{u \in GF(2)^n} \sum_{x \in GF(2)^n} (-1)^{g(x) \oplus g(y)} \underbrace{\sum_{y \in GF(2)^n} (-1)^{< u, x \oplus y >},}_{2^n \delta_x(y)} \\
&= 2^n \sum_{x \in GF(2)^n} (-1)^{2f(x)} = 2^{2n}, \qquad (2.41)
\end{aligned}$$

where

$$\delta_x(y) = \begin{cases} 1 & if \;\; y = x, \\ 0 & if \;\; y \neq x. \end{cases} \tag{2.42}$$

■

The following lemma is a similar result to Parseval's equation.

Lemma 41 $\sum_{u \in GF(2)^n} \Omega(\widehat{g})(u)\Omega(\widehat{g})(u \oplus v) = \begin{cases} 2^{2n} & if \;\; v = 0, \\ 0 & if \;\; v \neq 0. \end{cases}$

Proof. *The proof is straightforward and follows by lemma 31 and the fact* $\widehat{g}(w)^2 = 1$,

$$\begin{aligned}
\sum_{u \in GF(2)^n} \Omega(\widehat{g})(u)\Omega(\widehat{g})(u \oplus v) &= \sum_{u, w \in GF(2)^n} (-1)^{< u, w >} \widehat{g}(w) \sum_{u, w \in GF(2)^n} (-1)^{< u \oplus v, x >} \widehat{g}(x), \\
&= \sum_{u, w \in GF(2)^n} (-1)^{< v, x >} \widehat{g}(w) \widehat{g}(x) \sum_{u \in GF(2)^n} (-1)^{< u, (w \oplus x) >}, \\
&= 2^n \sum_{u, w \in GF(2)^n} (-1)^{< v, x >} \left(\widehat{g}(w)\right)^2 = 2^n \sum_{w \in GF(2)^n} (-1)^{< v, w >}. (2.43)
\end{aligned}$$

The case v=0, gives us a Parseval's equation. As mentioned earlier, we can start a relation between Walsh transform of the autocorrelation function and the square of the Walsh transform of the real-valued function. This fact is stated by the Wiener-Khintchine Theorem. ■

Theorem 42 *A Boolean function on* $GF(2)^n$ *satisfies*

$$\Omega(\widehat{r})(t) = \Omega(\widehat{g})^2(t), \tag{2.44}$$

for all $t \in GF(2)^n$*.*

Proof. *According to the definition of autocorrelation function, we obtain*

$$\begin{aligned}\Omega(\widehat{r})(t) &= \sum_{s\in GF(2)^n} (-1)^{<t,s>}\widehat{r}(s) = \sum_{s\in GF(2)^n}\left(\sum_{x\in GF(2)^n} (-1)^{g(x)\oplus g(x\oplus s)\oplus <t,s>}\right), \\ &= \sum_{x\in GF(2)^n}\left(\sum_{s\in GF(2)^n} (-1)^{g(x)\oplus g(x\oplus s)\oplus <t,s>}\right). \qquad (2.45)\end{aligned}$$

Since $GF(2)^n$ is invariant under any transformation, we may replace s by $x \oplus s$ in the second sum. Hence, we obtain

$$\begin{aligned}\Omega(\widehat{r})(t) &= \sum_{s\in GF(2)^n} (-1)^{<t,s>}\widehat{r}(s) = \sum_{x\in GF(2)^n}\left(\sum_{s\in GF(2)^n} (-1)^{g(x)\oplus g(s)\oplus <t,x\oplus s>}\right), \\ &= \left(\sum_{x\in GF(2)^n} (-1)^{g(x)\oplus <t,x>}\right)\left(\sum_{s\in GF(2)^n} (-1)^{g(s)\oplus <t,s>}\right), \\ &= \left(\sum_{x\in GF(2)^n} (-1)^{g(x)\oplus <t,x>}\right)^2 = \Omega(\widehat{g})^2(t). \qquad (2.46)\end{aligned}$$

■

Definition 43 *The spectral radius of a Boolean function $g : GF(2)^n \to GF(2)$ is defined by*

$$R_g = \max\{|\Omega(\widehat{g})(u)| : u \in GF(2)^n\} \qquad (2.47)$$

This definition provides a measure for linearity. Obviously, the linearity is upper bounded by $2^n \geq R_g$ by corollary 37. The upper bound is only attainable if g is affine.

Theorem 44 *For a Boolean function $g : GF(2)^n \to GF(2)$ the spectral radius is*

$$R_g \geq 2^{\frac{n}{2}}, \qquad (2.48)$$

and the equality is holds if and only if $\Omega(\widehat{g})^2 = 2^n$ is constant.

Theorem 45 *The nonlinearity of is determined by the Walsh transform of , that is,*

$$N_g = 2^{n-1} - \frac{1}{2}\max_{u\in GF(2)^n} |\Omega(\widehat{g})(u)|. \qquad (2.49)$$

Thus, it is possible to achieve high nonlinearity if the maximal Walsh-coefficient is of small value.

2.4 Correlation Immune Boolean Functions

Correlation immune functions were introduced by Siegenthaler [44] in order to protect some shift register based on stream ciphers against correlation attacks.

Definition 46 *[44] A Boolean function g in n variables is said to be correlation immune of order k, $1 \leq k \leq n$, if any fixed subset of k variables the probability that, given the value of $g(x)$, the k variables have any fixed set of values, is always 2^{-k}, no matter what the choice of the fixed set of k values is. In other words, g is correlation immune of order k if its values are statistically independent of any subset of input variables.*

We can formulate the definition of correlation immunity to an equivalent information theory condition. If the chosen subset of variables is $(x(i_1), x(i_2), ..., x(i_k))$ then the above definition of correlation immunity of order k is equivalent to the information theory condition that the information obtained about the values of $(x(i_1), x(i_2), ..., x(i_k))$ given is zero. Now we collect some useful equivalent conditions to correlation immunity of order 1 given by [3].

Lemma 47 *[54, Lemma 4.2]A function g in n variables is correlation immune of order 1 if and only if any of the following conditions holds.*

i. *If $\mathbf{supp}(g) = \{x \in GF(2)^n \mid g(x) = 1\}$, then for each $1 \leq i \leq n$, we have*
$\{x \in \mathbf{supp}(g)x_i = 1\} = \frac{\mathbf{supp}(g)}{2}$.

ii. *For each $1 \leq i \leq n$, $g(x) \oplus x_i$ is a balanced function.*

iii. *For each $1 \leq i \leq n$, $Prob(x_i = 1 \,|g(x) = 1\,) = \frac{1}{2}$ =$Prob(x_i = 0 \,|g(x) = 1\,)$.*

iv. *Let g_{0i} and g_{1i} denote the functions in $n-1$ variables obtained from g by setting $x_i = 0, 1$ respectively. Then for each , i=1,2,...,n, the functions g_{0i} and g_{1i} have the same Hamming weight.*

v. *All the Walsh transforms*

$$\Omega(\widehat{g})(u) = \sum_{x \in GF(2)^n} (-1)^{g(x) \oplus <u,x>}, \quad \mathbf{wt}(u) = 1, \tag{2.50}$$

are equal to zero.

vi. *For each i=1,2,...,n, $Prob(x_i = 1 \,|g(x) = 1\,) = \frac{1}{2}$ =$Prob(x_i = 0 \,|g(x) = 1\,) = \frac{\mathbf{wt}(g)}{2^n}$.*

Lemma 48 *A function g in n variables is correlation immune of order k, $1 \leq k \leq n$, if and only if all of the Walsh transforms*

$$\Omega(\widehat{g})(u) = \sum_{x \in GF(2)^n} (-1)^{g(x) \oplus <u,x>}, \quad 0 \leq \mathbf{wt}(u) \leq k, \tag{2.51}$$

are equal to zero.

Proof. The proof is based on the fact that the Walsh transforms is the cross correlation between and the linear functions . Let the $k-$vector y be defined by

$$y = (x(i_1), x(i_2), ..., x(i_k)) \,, \tag{2.52}$$

where $(x(i_1), x(i_2), ..., x(i_k))$ are variables in α_u. Then we focus on Walsh transforms in k variables of the conditional probability $Prob(y|x)$, where z is a possible value of g. By the definition of the expectation follows:

$$\sum_y Prob(y|z)(-1)^{<u,x>} = E\left[(-1)^{<u,x>}|g(x) = z\right] = E[(-1)^{<u,x>}] = \sum_y Prob(y)(-1)^{<u,x>}. \tag{2.53}$$

■

The equality follows by our correlation immunity hypothesis. Thus, $Prob(y|z)$ and $Prob(y)$ are identical since their Walsh transforms are identical. Consequently, the cross correlation between $g(x)$ and α_u is zero, which gives the statement. It follows from lemma 48 that the functions $g(x)$ and α_u are statistically independent if and only if the Walsh transforms if and only if $\Omega(\widehat{g})(u) = 0$.

We note that the original proof was given by Xiao and Massey [50]. Sarkar [42] gave another noteworthy proof which is based on linear algebra and combinatorics. Now, we obtain correlation value Now, we obtain the correlation value $c(g; \alpha_u)$. Therefore, we recall that the Hamming-distance between two Boolean functions $g, h : GF(2^n) \to \{1, -1\}$ is tied up with the cross correlation between g and h which is defined as

$$c(g, h) = \frac{\#\{x \in GF(2)^n|g(x) = h(x)\} - \#\{x \in GF(2)^n|g(x) \neq h(x)\}}{2^n}. \tag{2.54}$$

Now we use an arbitrary linear function α_u. Hence, we get

$$c(g, \alpha_u) = \Omega\left(\widehat{g}\right)(u)/2^n. \tag{2.55}$$

Thus, lemma 48 states that achieving correlation immunity for g is the same as getting zero correlation of with certain linear functions α_u. It is impossible to guarantee that g will not have a nonzero correlation with any linear function. This means we cannot achieve $c(g; \alpha_u) = 0$,for every u. This follows from the following lemma, which was first proven by Meier and Staffelbach [55].

Lemma 49 *For Boolean function g the total square correlation of g with the set of all linear functions is equal to one, that is*

$$\sum_{u \in GF(2)^n} c(g; \alpha_u)^2 = 1. \tag{2.56}$$

Proof. Proof. By equation (2.54), we have

$$\sum_{u \in GF(2)^n} c(g; \alpha_u)^2 = \sum_{u \in GF(2)^n} \Omega\left(\widehat{g}\right)(u)^2 / 2^{2n}, \tag{2.57}$$

then using result of the Parseval's corollary (2.39) and the statement follows immediately. As a result of lemma 49 and equation (2.54), we shift our focus to seeking those Boolean functions of which the largest possible value $|\Omega\left(\widehat{g}\right)(u)|$ of is as small as possible. These functions are the so-called perfect nonlinear functions which were introduced by Meier and Staffelbach [55]. ■

Definition 50 *A Boolean function in variables which is balanced and correlation immune of order k is said to be k−resilient function.*

Theorem 51 *Any Boolean function in n variables is k-resilient if and only if* $\Omega\left(\widehat{g}\right)(u) = 0$, *for all* $u \in GF(2^n)$ *such that* $\mathbf{wt}(u) \leq k$. *Equivalently, g is k-resilient if and only if* $\Omega\left(g\right)(u)$ *for all* $u \in GF(2^n)$ *such that* $0 < \mathbf{wt}(u) \leq k$.

Proof. See [8]. ■

Before we start to construct correlation immune functions we recall that a Boolean function cannot simultaneously have too many cryptographically desirable properties. In [44] Siegenthaler introduced a useful theorem which describes the relation between high order correlation immunity and high algebraic degree for a Boolean function, and we follow the more simple proof of Sarkar [42].

Theorem 52 *[54, Theorem 4.7]If g is a Boolean function in n variables, which is correlation immune of order k, then the degree of g is at most n-k. If g is also balanced and k<n-1, then the degree is at most n-k-1.*

Proof. See [54]. ■

2.5 Avalanche and Propagation Criterion

The avalanche effect states an appropriate property of cryptography. The avalanche consequence is obvious, when an input is altered to some extent the output changes meaningfully (e.g., half the output bits flip). The idea of avalanche was introduced by Horst Feistel which is based on the concept of Shannon's diffusion. The Strict Avalanche Criterion (SAC for short) was introduced by Webster and Tavares [102]. They write [102] "If a function is to satisfy the strict avalanche criterion, then each of its output bits should change with a probability of one half whenever a single input bit x is complemented to x'". The SAC is a useful property for Boolean functions in cryptographic applications. This means that if a Boolean function is satisfying the SAC, a small change in the input leads to a large change in the output (an avalanche effect). This property is essential in a cryptographic context due to the fact that we cannot infer its input from its output. In addition to SAC we study the Propagation Criterion (PC for short) which was introduced by Preneel et al. [39]. The mathematical expression for avalanche and SAC is defined as follows:

Definition 53 *A function $g : GF(2)^n \rightarrow GF(2)^m$ has the avalanche effect, if an average of $1/2$ of the output bits change whenever a single input bit is complemented i.e.*

$$\frac{1}{2^n} \sum_{u \in GF(2)^n} \mathbf{wt}(g(x^i) - g(x)) = \frac{m}{2}, \quad for\ all\ i = 1, 2, ..., n. \tag{2.58}$$

Definition 54 *A function $g : GF(2)^n \rightarrow GF(2)^m$ of n input bits into m output bits is said to be complete, if each output bit depends on each input bits, i.e. change whenever a single input bit is complemented i.e.*

$$\forall\ i = 1, 2, ..., n,\ j = 1, 2, ..., m,\ \exists\ x \in GF(2)^n \quad with \quad (g(x^i))_j \neq (g(x))_j. \tag{2.59}$$

If a cryptographic transformation is complete, then each ciphertext bit must depend on all of the output bits. Thus, if it were possible to find the simplest Boolean expression for each ciphertext bit in terms of the plaintext bits, each of those expressions would have to contain all of the plaintext bits if the function was complete. Alternatively, if there is at least one pair of n-bit plaintext vectors X and X_i that differ only in bit i, $g(X)$ and $g(X_i)$ differ at least in bit j for all $\{(i, j)|1 \leq i, j \leq n\}$ then the function g must be complete.

Definition 55 *A function $g : GF(2)^n \rightarrow GF(2)^m$ satisfies the strict avalanche criterion, if each output bit changes with a probability $1/2$ whenever a single input bit is complemented i.e.*

$$\forall\ i = 1, 2, ..., n,\ j = 1, 2, ..., m,\ Prob(g(x^i))_j \neq Prob(g(x))_j = \frac{1}{2}. \tag{2.60}$$

In the process of building these S-boxes, it was discovered that if an S-box is complete, or even perfect, its inverse function may not be complete. This could become important if these inverse functions are used in the decryption process, for it would be desirable for any changes in the ciphertext to affect all bits in the plaintext in a random fashion, especially if there is not much redundancy in the original plaintext. Complete cryptographic transformations with inverses which are complete are described as being two-way complete, and if the inverse is not complete the transformation is said to be only one-way complete.

Definition 56 *The dependence matrix of a function $g : GF(2)^n \rightarrow GF(2)^m$ is an $n \times m$ matrix A whose $(i, j)^{th}$ element a_{ij} denotes the number of inputs for which complementing the i^{th} input bit results in a* *change of the j^{th} output bit,*

$$a_{ij} = \#\{x \in GF(2)^n | \ \mathbf{wt}((g(x^i))_j - (g(x))_j\}, \quad for\ i = 1, 2, ..., n,\ and\ j = 1, 2, ..., m. \tag{2.61}$$

Definition 57 *The distance matrix of a function $g : GF(2)^n \to GF(2)^m$ is an $n \times (m+1)$ matrix B whose $(i, j)^{th}$ element b_{ij} denotes the number of inputs for which complementing i^{th} input bit results in a change of the j^{th} output bit, i.e.*

$$b_{ij} = \#\{x \in GF(2)^n |\}(g(x^i) - g(x)) = j\}, \quad for\ i = 1, 2, ..., n,\ and\ j = 1, 2, ..., m. \tag{2.62}$$

Definition 58 *For $g : GF(2)^n \to GF(2)$ and $a \in GF(2)^n$, $a \neq 0$, we defined the function by*

$$g_a(x) = g(x) \oplus g(x \oplus a), \tag{2.63}$$

where g_a is called the directional derivative of g in the direction of a.

Now we are able to express the SAC in connection with the directional derivative.

Lemma 59 *[54, Lemma 5.3]A Boolean function $g : GF(2)^n \to GF(2)$ satisfies SAC if and only if the function $g(x) \oplus g(x \oplus a)$ is balanced for every $a \in GF(2)^n$ with $a \neq 0$, Hamming-weight 1.*

Proof. We suppose that g fulfills the SAC, then exactly half of $a \in GF(2)^n$ with $a \neq 0$, satisfy $g(x) \neq g(x \oplus a)$ for every $a \in GF(2)^n$ with $\mathbf{wt}(a) = 1$. This means that

$$g(x) \oplus g(x \oplus a) = 1, \quad \text{for half the } a \in GF(2)^n, \tag{2.64}$$

$$g(x) \oplus g(x \oplus a) = 0, \quad \text{for half the } a \in GF(2)^n. \tag{2.65}$$

Summing up over $a \in GF(2)^n$ leads us to $\sum_{a \in GF(2)^n} g(x) \oplus g(x \oplus a) = 2^{n-1}$. So, $g(x) \oplus g(x \oplus a)$ is balanced. Lemma 59 provides a straightforward way to verify the SAC by computation of output values of g. ■

Definition 60 *The autocorrelation function of a Boolean function in n variables is defined as*

$$r_g(a) = \sum_{i=0}^{2^n-1} g(x_i) \oplus g(x_i \oplus a), \tag{2.66}$$

for all every $a \in GF(2)^n$.

The autocorrelation function is a simply the sum over all the values of the directional derivatives every $g(x) \oplus g(x \oplus a)$ as x runs through $GF(2)^n$. Now we are able to restate the Lemma 59 in terms of the autocorrelation function.

Lemma 61 *[54, Lemma 5.5]A Boolean function g in n variables is SAC if and only if the autocorrelation function $r_g(a)$ is equal to 2^{n-1} for all $a \in GF(2)^n$ with the Hamming-weight 1.*

2.6 The Strict Avalanche Criterion of Higher Order

In this section we study a generalization of the SAC defined by Forre [20], which she named the SAC of higher order.

Definition 62 *A Boolean function g in n variables is said to satisfy the SAC of order k $(SAC(k))$ if fixing any k in n bits in the input x results in a Boolean function in the remaining $n-k$ variables which satisfy the SAC, where $0 \leq k \leq n-2$.*

Lemma 63 *Suppose g is a Boolean function in $n > 2$ variables which satisfies the SAC of order k, $1 \leq k \leq n-2$. Then g also satisfies the SAC of order j for any j=0,1.*

Proof. See [50]. ■

2.7 The Propagation Criterion

This section generalizes the notion of the strict avalanche criterion to the propagation criterion.

Definition 64 *A Boolean function g in n variables is said to satisfy the propagation criterion of degree k ($PC(k)$ for short) if changing any i $(1 \leq i \leq k)$ of the n bits in the x input results in the output of the function being changed for exactly half of the 2^n vectors x.*

By the definition, we conclude that SAC is identical to $PC(1)$. The propagation criterion is strongly connected to properties of the autocorrelation function $r_g(a)$ as defined in definition 60.

Lemma 65 *A Boolean function in n variables satisfies $PC(1)$ if and only if all of the given values*

$$r_g(a) = \sum_{x \in GF(2)^n} g(x) \oplus g(x \oplus a), \qquad 1 \leq \mathbf{wt}(a) \leq k,$$

of the autocorrelation function are equal to 2^{n-1}.

Proof. From the definition of autocorrelation function $r_g(a)$,we have

$$Prob(g(x) \neq g(x \oplus a)) = \frac{r_g(a)}{2^n} = \frac{1}{2}. \qquad (2.67)$$

So the statement follows from the definition of the $PC(k)$. ■

The next lemma restates lemma 65 in terms of these directional derivatives 58.

Lemma 66 *A Boolean function g in n variables satisfies PC(k) if and only if all directional derivative* $g_a(x) = g(x) \oplus g(x \oplus a),\ 1 \leq \mathbf{wt}(a) \leq k,$ *are balanced functions.*

Proof. With lemma 65 and the definition of $PC(k)$ the statement follows immediately. ■

2.8 Properties of Nonlinearity

Nonlinearity is an important cryptographic criterion. It measures the ability of a cryptographic system using the functions to resist against being expressed as a linear set of equations.

The purpose of this section is to examine properties of nonlinearity and introduce some results concerning the upper and lower bound of nonlinearity. Furthermore, we observe ways to construct highly (balanced) nonlinear functions. A vast body of work has focused on nonlinearity. Thus, this section is mainly based on results from Seberry, Zhang and Zheng [58].

2.8.1 Bounds of Nonlinearity

In this section, we observe the upper bound of nonlinearity which is only attainable by bent functions. Moreover, we present some results about the lower bound of nonlinearity of a function obtained by concatenating sequences of functions. First, we phrase a lemma that is very useful in calculating nonlinearity of a function.

Lemma 67 *Let g and h be function on whose (1,-1)-sequences are* ξ_g *and* ξ_h*. Then the distance between g and h can be calculated by*

$$d(g,h) = 2^{n-1} - \frac{1}{2}\left\langle \xi_g, \xi_h \right\rangle. \tag{2.68}$$

Proof.

$$\begin{aligned}
\left\langle \xi_g, \xi_h \right\rangle &= \sum_{g(x)=h(x)} 1 - \sum_{g(x)\neq h(x)} 1, \\
&= 2^n - 2 \sum_{g(x)\neq h(x)}, \\
&= 2^n - 2d(g,h), \\
d(g,h) &= 2^{n-1} - \frac{1}{2}\left\langle \xi_g, \xi_h \right\rangle.
\end{aligned} \tag{2.69}$$

■

The valid question is whether we find an upper bound of nonlinearity.

Theorem 68 *For any function on* $GF(2)^n$ *the nonlinearity* N_g *of g satisfies* $N_g \leq 2^{n-1} - 2^{\frac{n}{2}-1}$.

2.9 Relation Between Cryptographic Properties

In this section, we will give some relation between cryptographic properties of Boolean functions. These relations are important while designing a new cryptographic Boolean function.

2.9.1 Relation Between Nonlinearity and Correlation Immunity

In this subsection, we obtain the relationship between nonlinearity and correlation immunity. First of all, we show two results given by Chee et al. [61] for arbitrary Boolean functions. Afterwards, we give a much stronger result using balanced functions.

Lemma 69 *Let g be a Boolean function in n variables and we define*

$$\eta(g) = |\{w \in GF(2)^n : \Omega(\widehat{g})(w) \neq 0\}|. \tag{2.70}$$

Then $N_g \leq 2^{n-1} - 2^{n-1}\eta(g)^{-\frac{1}{2}}$.

Proof. By Parseval's equation (see Corollary 40) we have

$$2^{2n} = \sum_{w \in GF(2)^n} \Omega(\widehat{g})(w)^2 \leq \eta(g) \max_{w \in GF(2)^n} |\Omega(\widehat{g})|^2. \tag{2.71}$$

So it follows that $\max_{w \in GF(2)^n} |\Omega(\widehat{g})|^2 \geq 2^n \eta(g)^{-\frac{1}{2}}$. Using theorem 45, we have

$$N_g = 2^{n-1} - \frac{1}{2} \max_{w \in GF(2)^n} |\Omega(\widehat{g})| \leq 2^{n-1} - 2^{n-1}\eta(g)^{-\frac{1}{2}}. \tag{2.72}$$

■

Lemma 70 *If g is any Boolean function in n variables which is correlation immune of order k and*

$$\mu(n,k) = 2^n - \sum_{i=0}^{k} \left(\begin{array}{c} n \\ i \end{array} \right), \tag{2.73}$$

then $N_g \leq 2^{n-1} - 2^{n-1}\mu(n,k)^{-\frac{1}{2}}$.

Proof. Since is correlation immune of order , Lemma 48 implies that

$$\begin{aligned} \eta(g) &= 2^n - |\{w \in GF(2)^n : \Omega(\widehat{g})(w) = 0\}|, \\ &\leq 2^n - |\{w \in GF(2)^n : 1 \leq \mathbf{wt}(a) \leq k\}| = \mu(n,k). \end{aligned} \tag{2.74}$$

Thus, N$_g \leq 2^{n-1} - 2^{n-1}\mu(n,k)^{-\frac{1}{2}}$,follows from Lemma 69. ■

Lemma 71 *For every Boolean function g on GF(2)^n, we have*

$$(r_g(a_0), r_g(a_1), r_g(a_2), ..., r_g(a_{2^n-1})) .H_n = \left(\left\langle \xi_g, h_0 \right\rangle^2, \left\langle \xi_g, h_1 \right\rangle^2, \left\langle \xi_g, h_2 \right\rangle^2, ..., \left\langle \xi_g, h_{2^n-1} \right\rangle^2\right), \quad (2.75)$$

where ξ denotes the (1,–1)- sequence of g and h_i is the i^{th} row of H_n, and a_i is defined in definition 1, $i = 0, 1, 2, ..., 2^n - 1$.

In the next theorem, we use the property of balancedness to obtain a kind of trade-off between nonlinearity and correlation immunity. This theorem is given by Tarannikov [46] and we follow his proof.

Theorem 72 *Let g be a balanced and correlation immune function of order k, $k \leq n - 2$, then $N_g \leq$* $2^{n-1} - 2^{k+1}$.

Theorem 73 *Let g be a balanced Boolean function in n variables which is correlation immune of order $k \leq n - 2$. Then equality is possible in theorem 72 only if has its maximum possible degree $n - k - 1$. If deg$(g) < n - k - 1$, then $N_g \leq 2^{n-1} - 2^{k+2}$.*

Proof. We use the same subfunction h as in proof of theorem 72. So $g(x|x_{i(1)} = a_1, ..., x_{i(k+1)} = a_{k+1}) = h$ which has $\mathbf{wt}(h) = w < 2^{n-k-2}$. By theorem, we know that $\deg(h) \leq \deg(g) \leq n-k-1$. Thus, equality is possible in theorem 72. If $\deg(g) \leq n-k-1$ then the subfunction h must have even Hamming-weight because h is a function in $n - k - 1$ variables. Therefore, we have $w \leq 2^{n-k-2} - 2$. By the same proof of theorem 72, it follows that $N_g \leq 2^{n-1} - 2^{k+2}$. ■

2.9.2 Relationship Between Nonlinearity and the Propagation Criterion

In this subsection, we observe the relationship between nonlinearity and the propagation criterion. Finally, we have presented a relationship between nonlinearity and the propagation criterion given by Zhang and Zheng [62]. The following theorem presents the main result about the relationship between non-linearity and the propagation criterion.

Theorem 74 *Let g be a Boolean function on GF(2)^n. Further g satisfies the propagation criterion of degree k . Then*

(i). The nonlinearity N_g of g is $N_g \geq 2^{n-1} - 2^{n-1-\frac{k}{2}}$.

(ii). The equality in (i) holds if and only if one of the following two conditions holds:

(a). $k = n - 1$, n is odd.

(b). $k = n$, g is bent and n is even.

2.10 Some Special Boolean Functions

We now discuss three special types of Boolean functions, and the identifying properties which enable them to be classified as such.

2.10.1 Bent Functions

A particular class of Boolean functions exhibiting unique characteristics was first reported by Rothaus in [41]. These functions were referred to in that paper as bent functions. Bent functions were later called perfect nonlinear [31] in light of their optimal distance to linear structures.

Bent functions exist only in the space of even dimensional Boolean functions. It is not possible for a Boolean function to exist that satisfies all the necessary characteristics to be considered bent when the space is odd dimensional.

The Walsh Hadamard spectrum of a Boolean function is a two-valued spectrum and consists entirely of $\pm 2^{n/2}$ values. Thus, the Walsh Hadamard spectrum is flat. It follows by definition that the nonlinearity of a bent function will be $(2^n - 2^{n/2})/2$. As Parseval's Theorem must hold, clearly this is the maximum achievable nonlinearity for n-dimensional Boolean functions (n even). This indicates that a bent function is at maximum distance to linear structures. Further, as there can be no zero-valued entries in the Walsh Hadamard spectrum, bent functions do not exhibit any order of correlation immunity.

The autocorrelation vector of an n-variable bent Boolean function (n even) takes the form $\widehat{r}(\alpha) = \{2^n, 0, 0, ..., 0\}$. The first entry always has the value 2^n and all other entries are 0. Thus, bent functions satisfy propagation criteria of degree n, $PC(n)$, and exhibit perfect diffusion with respect to output uniformity given shifts in the input of a bent function.

Although bent functions exhibit cryptographically optimal properties in terms of maximal nonlinearity and perfect (minimal) autocorrelation, n-variable bent functions have a Hamming weight of $2^{n-1} \pm 2^{n/2-1}$. This indicates a bias from balance of constant magnitude $2^{n/2-1}$; bent functions are never balanced. Furthermore, all n-variable bent functions have algebraic degree are cryptographically undesirable for bent functions to be of direct practical use. Various techniques for the construction of bent functions have been proposed in the literature. Some examples include [30], [57], [58], [59] and [60].

2.10.2 Semi-Bent Functions

The cryptographic limitations of bent functions discussed above (unbalanced, low algebraic degree) prevent bent functions from being useful cryptographically. Semi-bent functions were introduced by Chee et al. in [33] [61]. These semi-bent functions attempt to retain the desirable characteristics of bent functions, namely high nonlinearity and zero autocorrelation, whilst ensuring balance.

A semi-bent function, $g(x)$, is an odd-demensional Boolean function constructed by concatenating a bent function, $h(x)$, to the same bent function, $h(x)$, that has had an affine transformation applied to its input and its output complemented.

Definition 75 *(From [61]) Let $g(x)$ be an n-variable semi-bent Boolean function (n odd) and $h(x)$ be and $(n-1)$-variable bent function. Then $g(x)$ is of the form*

$$g(x) = h(x) \parallel (h(Ax \oplus b) \oplus 1). \tag{2.76}$$

A semi-bent function, $g(x)$, constructed in this manner is always balanced. The nonlinearity of an n-variable semi-bent function is $2^{n-1} - 2^{\frac{n-1}{2}}$.

The correlation coefficients between an n-variable semi-bent function (n odd) and the set of all n-variable linear functions always take one of the values in the set $\{0, \pm 2^{\frac{1-n}{2}}\}$. For an n-variable semi-bent function, $g(x)$, $\#(C(g,l) = 0) = 2^{n-1}$ and $\#(C(g,l) = \pm 2^{\frac{1-n}{2}}) = 2^{n-1}$ for all l in the set of n-variable linear Boolean functions. The former represents no correlation between $g(x)$ and half of all n-variable linear functions. The latter indicates uniform correlation to the other half of the set of all linear functions.

An n-variable semi-bent function (n odd) of degree $(n-1)$ also satisfies propagation criteria of degree n, $PC(n)$. The Strict uncorrelated Criterion as introduced and defined by Chee et. al. in [61] may also be satisfied by pairs of semi-bent functions under certain conditions. Thus, semi-bent functions represent a useful grouping of odd-dimensional Boolean functions with a number of good combined cryptographic properties.

2.10.3 Plateaued Functions

A class of n-variable Boolean functions (n both odd and even) were introduced in [62] and termed "plateaued" functions. Before we outline the main characteristics of plateaued functions, we present the definition form [62].

Definition 76 *(From [62]) Let $g(x)$ be an n-variable Boolean function with Walsh Hadamard transform vector, $\widehat{\Omega}(\omega)$. Let $k = \{\#\omega \mid \widehat{\Omega}(\omega) \neq 0\}$. Then $g(x)$ is a plateaued function if for all $\omega \in GF(2)^n$, the square of the elements of $\widehat{\Omega}(\omega)$, $\left(\widehat{\Omega}(\omega)\right)^2 \in \{0, 2^{2n-1}\}$ for some even t such that $k = 2^t$ $(0 \leq t \leq n)$. $g(x)$ may also be known as a plateaued function of order t.*

Thus, a plateaued function of order t (if $t \neq n$) has a three-valued Walsh Hadamard spectrum. The nonlinearity of n-variable plateaued functions is $2^{n-1} - 2^{n-\frac{t}{2}-1}$. The higher the order t of a plateaued function, the greater the nonlinearity of the function. For n even, plateaued functions of order n are the bent functions. The plateaued functions of order 0 correspond to the affine functions. From Definition 76, the number of zeros in the Walsh Hadamard spectrum of a t^{th}-order plateaued function is $2^n - 2^t$. Therefore, there exist balanced and correlation immune n-variable plateaued functions.

It is proposed in [62] that the algebraic degree of a plateaued function $g(x)$ of order t is such that $\deg(g) \leq t/2 + 1$. Consequently, this would mean that plateaued functions do not exhibit high algebraic degree. The sum-of-square indicator of an n-variable plateaued function is equal to $2^{3n}/k = 2^{3n-t}$. As expected from the previous paragraph and the discussion of section, the sum-of-square indicator will be low for large t.

A subset of plateaued functions is the set of partially-bent functions introduced in [10]. Unlike partially-bent functions, which always possess non-zero linear structures, plateaued functions with no non-zero linear structures exist. The reader is referred to [10] for a description of partially-bent functions. Example constructions for plateaued functions can be found in [62] and [63].

We have seen that plateaued functions may possess desirable properties such as balance, correlation immunity, high nonlinearity and low sum-of-square indicator. As with all Boolean functions, the extent to which combinations of certain properties will be exhibited together are determined by the complementary or opposing nature of their relationships. Unlike bent and semi-bent Boolean functions, plateaued functions may have an even or odd number of input variables. We have discussed three special types of Boolean functions of interest in this thesis. The reader should be aware that there are other special Boolean functions which are not discussed, such as partially-bent functions [10].

2.11 S-Box Theory

In this section we now turn our discussions to the area of substitution boxes (S-boxes). The basic definitions of S-box theory are provided to support the research work performed in this thesis. Also in this section, a review of relevant cryptographic properties as applied to S-boxes, is provided.

2.11.1 S-Box Definitions and Types

A natural progression from the theory of single output Boolean functions is the extension of that theory to multiple output Boolean functions, collectively referred to as an S-box. The relationship between the input and output bits in terms of dimension and uniqueness gives rise to various types of S-boxes. We list below several necessary S-box definitions, together with a brief description of some S-box types of interest to this research.

An $n \times m$ substitution box (S-box) is a mapping from n input bits to m output bits, $S : GF(2)^n \rightarrow GF(2)^m$. The output vector $S(x) = (s_1, s_2, ..., s_m)$ can be decomposed into m component functions $S_i : GF(2)^n \rightarrow GF(2),\ i = 1, 2, ..., m$. There are 2^n inputs and 2^m possible outputs for an $n \times m$ S-box. Often considered as a look-up table, an $n \times m$ S-box, S, is normally symbolized as a matrix of size $2^n \times m$, indexed as $S_{[i]}$ $(0 \leq i \leq 2^n - 1)$ each an $m-$bit entry. There are, generally speaking, three types of S-boxes: Straight, compressed and expansion S-boxes.

A straight $n \times m$ S-box with $n = m$ (takes in a given number of bits and puts out the same number of bits) may either contain distinct entries where each input is mapped to a distinct output OR repeat S-box entries where multiple inputs may be mapped to the same output and all possible outputs are not represented in the S-box. An $n \times m$ S-box which is both injective and surjective is known as a bijective S-box. That is, each input maps to a distinct output entry and all possible outputs are present in the S-box. Bijective S-boxes may only exist when $n = m$ and are also called reversible since there must also exist a mapping from each distinct output entry to its corresponding input. This is the design approached used with the Rijndael cipher.

A compression $n \times m$ S-box $n > m$ with puts out fewer bits than it takes in. A good example of this is the S-box used in DES. In the case of DES, each S-box takes in 6 bits but only outputs 4 bits. A expansion $n \times m$ S-box with $n < m$ puts out more bits than it takes in. A regular $n \times m$ S-box is one which has each of its possible 2^m output appearing an equal number of times in the S-box. Thus, each of the possible output entries appears a total number of 2^{n-m} times in the S-box. All single output Boolean functions comprising a regular S-box are balanced, as are all linear combinations of these functions. Regular $n \times m$ S-boxes are balanced S-boxes and may only exist when $n \geq m$. An $n \times m$ S-box ($n \geq 2m$ and n is even) is said to be bent if every linear combination of its component Boolean functions is a bent function.

There are issues associated with both compression and expansion S-boxes. The first issue is reversibility, or decryption. Since either type of S-box alters the total number of bits, reversing the process is difficult. The second issue is a loss of information, particularly with compression S-boxes. In the case of DES, prior to the S-box, certain bits are replicated. Thus what is lost in the compression step are duplicate bits and no information is lost. In general working with either compression or expansion S-boxes will introduce significant complexities in your S-box design. Therefore straight S-boxes are far more common.

2.11.2 Cryptographic Properties of S-Boxes

While many of the Boolean function properties discussed in previous sections have conceptual equivalences when applied to S-boxes, there are fundamental differences in the manner by which these properties are derived. As an S-box is comprised of a number of component Boolean functions, it is important to observe that when considering the cryptographic properties of an S-box, it is not sufficient to consider the cryptographic properties of the component Boolean functions individually. Rather, it is also necessary to consider the cryptographic properties of all the linear combinations of the component functions. This is illustrated in the following selection of relevant S-box properties.

An $n \times m$ S-box which is balanced is one whose component Boolean functions and their linear combinations are all balanced. Because of this balance, there does not exist an exploitable bias in that the equally likely number of output bits over all output vector combinations ensures that an attacker is unable to trivially approximate the functions or the output.

The well-known concept of confusion due to Shannon [5] is described as a method for ensuring that in a cipher system a complex relationship exists between the ciphertext and the key material.

This notion has been extrapolated to mean that a significant reliance on some form of substitution is required as a source of this confusion. The confusion in a cipher system is achieved through the use of nonlinear components. As expected, substitution boxes tend to provide the main source of nonlinearity to cryptographic cipher systems. We now define the measure of nonlinearity for an $n \times m$ S-box.

Definition 77 *The nonlinearity of an $n \times m$ S-box S, denoted by $N_{S_{n,m}}$ is defined as the minimum nonlinearity of each of its component output Boolean functions and their linear combinations. Let $S = (s_1, s_2, ..., s_m)$ where s_i $(i = 1, ..., m)$ are n-variable Boolean functions. Let h_i be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the functions s_i). Then the nonlinearity of S can be expressed as follows:*

$$N_{S_{n,m}} = \min_{h}\{N_{S_{n,m}}(h_j)\} \ (j = 1, ..., 2^m - 1). \tag{2.77}$$

Clearly, as n and m increase, the task of merely computing the nonlinearity value of an $n \times m$ S-box quickly becomes computationally infeasible. The importance of this property for the security of cipher systems becomes evident in the next section when we discuss some of the effective cryptanalytic attacks which exist.

The algebraic degree of an S-box (and similarly a Boolean function) is desired to be as high as possible in order to resist a cryptanalytic attack known as low order approximation [29]. The measure of S-box degree is defined below:

Definition 78 *Let $S = (s_1, s_2, ..., s_m)$ be an $n \times m$ S-box where s_i $(i = 1, ..., m)$ are n-variable Boolean functions. Let h_j be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the functions h_i). Then the algebraic degree of S, denoted by $\deg(S_{n,m})$, is defined as*

$$\deg(S_{n,m}) = \min_{h}\{\deg(h_j)\} \ (j = 1, ..., 2^m - 1). \tag{2.78}$$

A companion concept to confusion, called diffusion, was also proposed by Shannon in [5]. Therein it is described as the method by which the data redundancy in a cipher is spread throughout the entire (or large portion of the) data in an effort to reduce the probability of discovering part or all of its statistical structure. Diffusion has long been lined to the avalanche characteristics of a cipher system and, in particular, is achieved by using cipher components which exhibit good avalanche characteristics. In order to measure these characteristics for $n \times m$ S-boxes we require the following definitions:

Definition 79 *Let $S = (s_1, s_2, ..., s_m)$ be an $n \times m$ S-box where s_i $(i = 1, ..., m)$ are n-variable Boolean functions. Let h_j be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the functions s_i), each with autocorrelation function, $\widehat{r}_{h_j}(a)$. Then the maximum absolute autocorrelation value of S is defined as:*

$$\left|AC_{S_{n,m}}\right|_{\max} = \max_{h}\left|\widehat{r}_{h_j}(a)\right|, \tag{2.79}$$

with $a \in \{1, ..., 2^n - 1\}$ and $(j = 1, ..., 2^m - 1)$.

Definition 80 *Let $S = (s_1, s_2, ..., s_m)$ be an $n \times m$ S-box where s_i $(i = 1, ..., m)$ and n-variable Boolean functions. Let h_j be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the function s_i). Then S is said to satisfy strict avalanche criterion (SAC) if every h_j $(j = 1, ..., 2^m - 1)$ satisfies SAC.*

Definition 81 *Let $S = (s_1, s_2, ..., s_m)$ be an $n \times m$ S-box where s_i $(i = 1, ..., m)$ are n-variable Boolean functions. Let h_j be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the functions s_i). Then S is said to satisfy propagation criteria of order k, $PC(k)$, if every h_j $(j = 1, ..., 2^m - 1)$ satisfies $PC(k)$.*

The next two definitions outline the way in which, respectively, the correlation immunity and resilience of an S-box are determined.

Definition 82 *Let $S = (s_1, s_2, ..., s_m)$ be an $n \times m$ S-box where s_i $(i = 1, ..., m)$ are n-variable Boolean functions. Let h_j be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the functions s_i). Then S is a $CI(t)$ S-box if all h_j $(j = 1, ..., 2^m - 1)$ are $CI(t)$ Boolean functions.*

Definition 83 *Let $S = (s_1, s_2, ..., s_m)$ be an $n \times m$ S-box where s_i $(i = 1, ..., m)$ are n-variable Boolean functions. Let h_j be the set of linear combinations of s_i $(i = 1, ..., m)$ (which includes the functions s_i). Then S is a t-resilient S-box if all h_j $(j = 1, ..., 2^m - 1)$ are t-resilient Boolean functions.*

2.12 Bit Independent Criterion

Webster and Tavares in 1985, introduced another criterion, called bit independent criterion (BIC) for S-Boxes [48]. This property states that the output bits j and k should alter independently, when any single input bit i is reversed, for all i, j and $k \in (1, 2, ..., n)$. This criterion appears to strengthen the effectiveness of the confusion function. To illustrate the bit independent concept, one requires the correlation coefficient between j^{th} and k^{th} components of the output difference string. The bit independence corresponding to the effect of the i^{th} input bit change on the j^{th} and k^{th} bits of is B^{e_i}:

$$BIC(b_j, b_k) = \max_{1 \le i \le n} |\mathbf{corr}(b_j^{e_i}, b_k^{e_i})|. \tag{2.80}$$

The bit independent criterion (BIC) parameter for the S-box function $h : GF(2)^n \to GF(2)^n$, is then defined as follows:

$$BIC(h) = \max_{\substack{1 \le j,k \le n \\ j \ne k}} BIC(b_j, b_k), \tag{2.81}$$

which shows how close is satisfying the BIC [48].

2.13 Linear and Differential Cryptanalysis of S-boxes

Linear cryptanalysis was introduced at Eurocrypt conference in 1993 by M. Matsui as a theoretical attack on the Data Encryption Standard (DES) [65], [15] and later successfully used in the practical cryptanalysis of DES [29]. Linear cryptanalysis works on the principle of finding "high probability occurrences of linear expressions involving plaintext bits, ciphertext bits (actually we shall use bits from the 2nd last round output), and subkey bits" [66]. It is a known plaintext attack in which a large number of plaintext-ciphertext pairs are used to determine the value of key bits [66].

Differential cryptanalysis was first presented at Crypto conference in 1990 by E. Biham and A. Shamir as an attack on DES [67]. Heys [66] describes the main principle: "Differential cryptanalysis exploits the high probability of certain occurrences of plaintext differences and differences into the last round of the cipher". It is a chosen plaintext attack, that means plaintext can be selected and output subsequently calculated in order to derive the key [68]. In this section XOR distribution, linear and differential probability are defined.

Definition 84 *For an vector Boolean function (S-boxes) $g : GF(2)^n \to GF(2)^n$, the XOR table has a size of $2^n \times 2^n$, with its rows and columns indexed by $0, 1, 2, ..., 2^n - 1$. Position (i, j) in the XOR table contains the number of input vectors:*

$$|\{P \in GF(2)^n : g(P) \oplus g(P \oplus \tau_i) = \tau_j\}|, \tag{2.82}$$

such that $0 \leq i, j \leq 2^n - 1, \tau_i$ and τ_j are n-bit binary representations of indices i and j. P is the input vector, g corresponds to the cryptographic function of the S-box, and the pair (i, j) is called an input/output XOR pair. Differential cryptanalysis exploits such XOR pairs with large XOR table entries. A cipher can be secured against differential cryptanalysis by selecting S-boxes with low XOR table entries, ideally 0 or 2 (the only exception is the entry $(0, 0)$ which has the value of 2^n). The sum of the XOR table entries on each row is equal to 2^n, which is the total number of input vector pairs $(P, P \oplus \tau_i\ )$ [69].

Definition 85 *For a given vector Boolean function $g : GF(2)^n \to GF(2)^m$ it is defined the linear approximation table which elements are*

$$LAT_g(a, b) = \#\{x \in GF(2)^n | a.x = b.g(x)\} - 2^{n-1}, \tag{2.83}$$

where $a \in GF(2)^n,\ b \in GF(2)^m \backslash \{0\}$.

Lemma 86 *For a given vector Boolean function $g : GF(2)^n \to GF(2)^m$ it is defined the linear approximation table which elements are*

$$LAT_g(a, b) = 2^{n-1} - d(a.x, b.g), \tag{2.84}$$

where $a \in GF(2)^n,\ b \in GF(2)^m \backslash \{0\}$.

Proof. By the definition of LAT, we have

$$\begin{aligned} LAT_g(a,b) &= \#\{x \in GF(2)^n | a.x = b.g(x)\} - 2^{n-1}, \\ &= 2^n - \#\{x \in GF(2)^n | a.x \neq b.g(x)\} - 2^{n-1}, \\ &= 2^{n-1} - d(a.x, b.g). \end{aligned} \tag{2.85}$$

■

Lemma 87 *For a given vector Boolean function $g : GF(2)^n \rightarrow GF(2)^m$ one has*

$$N_g = 2^{n-1} - \max_{a,b} |LAT_g(a,b)|, \tag{2.86}$$

where $a \in GF(2)^n,\ b \in GF(2)^m \backslash \{0\}$.

Definition 88 *For any given $\Delta_x,\ \Delta_y\ ,\ \Gamma_x,\ \Gamma_y\ \in GF(2)^n$,the linear and differential approximation probabilities for each vector Boolean function (S-box) are defined as:*

$$LP^{S_i}(\Gamma_y \quad \rightarrow \quad \Gamma_x) = \left(2 \times \frac{\#\{x \in GF(2)^n | x\Gamma_x = S_i(x)\Gamma_y\}}{2^n} - 1\right), \tag{2.87}$$

$$DP^{S_i}(\Delta_x \quad \rightarrow \quad \Delta_y) = \left(\frac{\#\{x \in GF(2)^n | S_i(x) \oplus S_i(x \oplus \Delta_x) = \Delta_y\}}{2^n}\right), \tag{2.88}$$

where $x\Gamma_x$, denotes the parity (0 or 1) of the bitwise product of x and Γ_x.

Definition 89 *The maximum linear and differential approximation probabilities of vector Boolean function (S-boxes) are defined as:*

$$p = \max_i \max_{\Gamma_x, \Gamma_y} LP^{S_i}(\Gamma_y \rightarrow \Gamma_x), \tag{2.89}$$

$$q = \max_i \max_{\Delta_x, \Delta_y} DP^{S_i}(\Delta_x \rightarrow \Delta_y). \tag{2.90}$$

2.14 Conclusion

In this chapter, we have defined the relevant supporting theory of both Boolean functions and substitution boxes. In particular, we have provided numerous long established definitions and theorems for various aspects of the theory. The necessary cryptographic properties which are used to analyze the strength of single and multiple output functions have also been defined and discussed, as have the inter-relations between pairs of selected properties.

Chapter 3

Introduction to Information Security Systems Primitives

This chapter is devoted to introducing the preliminaries related to information security systems to be discussed in this thesis. We explicitly define the basics of the cryptographic, watermarking and steganographic primitives which will be helpful in subsequent chapters.

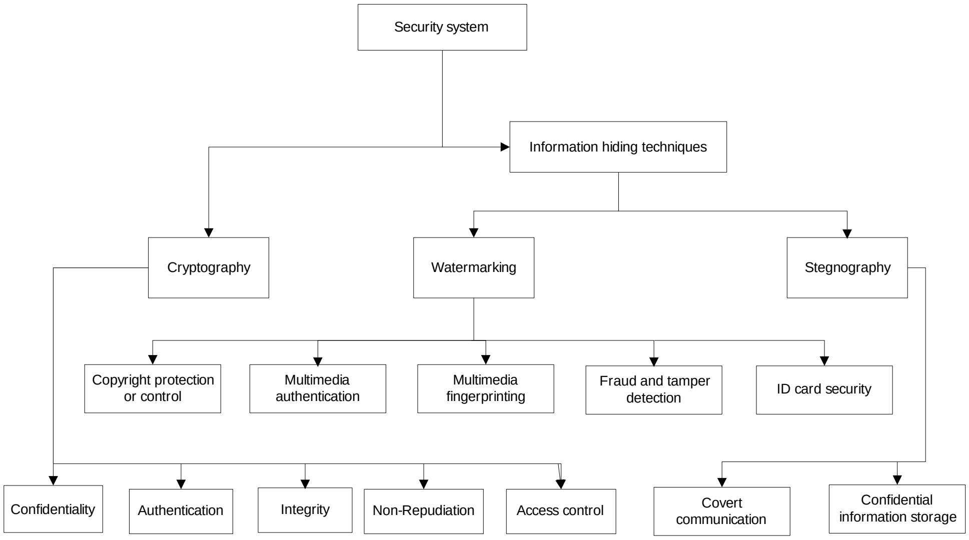

Security system
Information hiding techniques
Cryptography
Watermarking
Stegnography
Copyright protection or control
Multimedia authentication
Multimedia fingerprinting
Fraud and tamper detection
ID card security
Confidentiality
Authentication
Integrity
Non-Repudiation
Access control
Covert communication
Confidential information storage

Fig. 3.1: Classification of information security techniques and its applications.

3.1 Cryptology

Cryptology is the science of information security. The word is derived from the Greek kryptos, meaning hidden. Cryptology is the study of "secret writing." Modern cryptology combines the studies of computer science and mathematics for the purpose of encoding information to ensure that data is secure.

3.1.1 Classification of Cryptology

The cryptology is further classified into two branches cryptography and cryptanalysis. The term cryptography refers to the art or science of designing cryptosystems (to be defined shortly), while cryptanalysis refers to the science or art of breaking them. Although cryptology is the name given to the field that includes both of these, we will generally follow the common practice (even among many professionals and researchers in the field) of using the term cryptography" interchangeably with \cryptology" to refer to the making and breaking of cryptosystems. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

3.1.2 Basics Terminology of Cryptography

Plain Text

The original message that the person wishes to communicate with the other is defined as Plain Text. In cryptography the actual message that has to be send to the other end is given a special name as Plain Text.

Cipher Text

The message that cannot be understood by anyone or meaningless message is what we call as Cipher Text. In Cryptography the original message is transformed into non-readable message before the transmission of actual message.

Ciphers

A cipher encrypts a single letter or group of letter as a unit, regardless of meaning.

Codes

A code encodes a word or phrase at a time usually in a fixed way (no keys).

Encryption

A process of converting plain-text into cipher-text is called as encryption. Cryptography uses the encryption technique to send confidential messages through an insecure channel. The process of encryption requires two things- an encryption algorithm and a key. An encryption algorithm means the technique that has been used in encryption. Encryption takes place at the sender side.

Decryption

A reverse process of encryption is called as decryption. It is a process of converting cipher-text into plain-text. Cryptography uses the decryption technique at the receiver side to obtain the original message from non-readable message (Cipher Text). The process of decryption requires two things- a decryption algorithm and a key. A decryption algorithm means the technique that has been used in decryption. Generally the encryption and decryption algorithm are same.

Key

A Key is a numeric or alpha numeric text or may be a special symbol. The key is used at the time of encryption takes place on the plain-text and at the time of decryption take place on the cipher-text. The selection of key in cryptography is very important since the security of encryption algorithm depends directly on it.

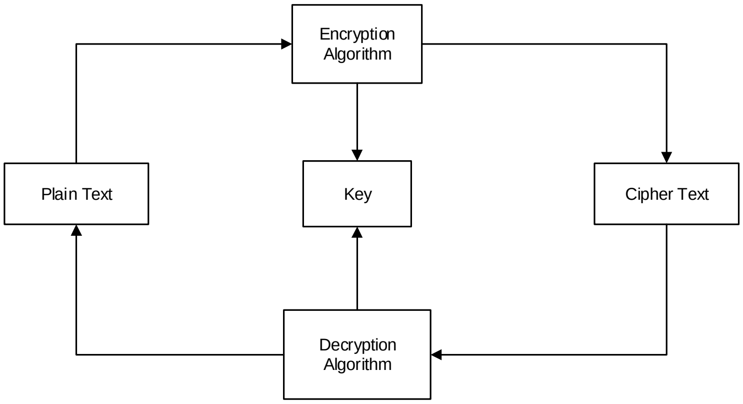


Fig. 3.2: Block diagram for encryption and decryption.

3.2 Purpose of Cryptography

Cryptography provides a number of security goals to ensure the privacy of data, non- alteration of data and so on. Due to the great security advantages of cryptography it is widely used today. Following are the various goals of cryptography.

3.2.1 Confidentiality

Information in computer is transmitted and has to be accessed only by the authorized party and not by anyone else.

3.2.2 Authentication

The information received by any system has to check the identity of the sender that whether the information is arriving from an authorized person or a false identity.

3.2.3 Integrity

Only the authorized party is allowed to modify the transmitted information. No one in between the sender and receiver are allowed to alter the given message.

3.2.4 Non Repudiation

Ensures that neither the sender, nor the receiver of message should be able to deny the transmission.

3.2.5 Access Control

Only the authorized parties are able to access the given information.

3.3 Classification of Cryptography

Encryption algorithms can be classified into two broad categories- Symmetric and Asymmetric key encryption.

3.3.1 Symmetric Encryption

In symmetric cryptography the key used for encryption is similar to the key used in decryption. Thus the key distribution has to be made prior to the transmission of information. The key plays a very important role in symmetric cryptography since their security directly depends on the nature of key i.e. the key length etc. There are various symmetric key algorithms such as DES, TRIPLE DES, AES, RC4, RC6,BLOWFISH [2]. The symmetric algorithms are of two types namely block ciphers and stream ciphers which are defined as follows [71].

Block Ciphers

A block cipher is a function which maps n bit plaintext blocks to n bit cipher-text blocks; n is called the block length. It may be viewed as a simple substitution cipher with large character size. The function is parameterized by a k-bit key K, taking values from a subset K (the key space) of these to fall k-bit vectors V_k. It is generally assumed that the key is chosen at random. Use of plaintext and ciphertext blocks of equal size avoids data expansion.

Definition 90 *An n bit block cipher is a function $E{:}V_n \times K \rightarrow V_n$ such that for each key $k \in K$, $E(P;k)$ is an invertible mapping (the encryption function for k) from V_n to V_n, can be written as $E_k(P)$. The inverse mapping is the decryption function, denoted by $D_k(C)$. $C = E_k(P)$ denote that cipher text C results from encrypting plaintext P under k[71].*

Stream Ciphers

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate, and in some cases mandatory (e.g., in some telecommunications applications), when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable. Stream ciphers are commonly classified as being synchronous or self-synchronizing.

Synchronous Stream Ciphers A synchronous stream cipher is one in which the key stream is generated independently of the plaintext message and of the cipher text. The encryption process of a synchronous stream cipher can be described by the equations [71]:

$$\beta_{i+1} = f(\beta_i, K), \tag{3.1}$$

$$\alpha_i = g(\beta_i, K), \tag{3.2}$$

$$c_i = h(a_i, m_i), \tag{3.3}$$

where β_0 is the initial state and may be determined from the key K, f is the next-state function, g is the function which produces the key stream α, and h is the output function which combines the key stream and plaintext m to produce cipher-text c.

Self-synchronizing Stream Ciphers A self-synchronizing or asynchronous stream cipher is one in which the key-stream is generated as a function of the key and a fixed number of previous cipher text digits. The encryption function of a self-synchronizing stream cipher can be described by the equations [71]:

$$\beta_{i+1} = (c_{i-t}, c_{i-t+1}, ..., c_{i-1}), \tag{3.4}$$

$$\alpha_i = f(\beta_i, K), \tag{3.5}$$

$$c_i = g(a_i, m_i), \tag{3.6}$$

where $\beta_0 = (c_{-t}, c_{-t+1}, ..., c_{-1})$ is the (non-secret) initial state, K is the key, f is the function which produces the key stream α, and g is the output function which combines the key stream and plaintext m to produce cipher-text c.

3.3.2 Asymmetric Encryption

Asymmetric cryptography or public-key cryptography is cryptography in which a pair of keys is used to encrypt and decrypt a message so that it arrives securely. Initially, a network user receives a public and private key pair from a certificate authority. Any other user who wants to send an encrypted message can get the intended recipient's public key from a public directory. They use this key to encrypt the message, and they send it to the recipient. When the recipient gets the message, they decrypt it with their private key, which no one else should have access to.

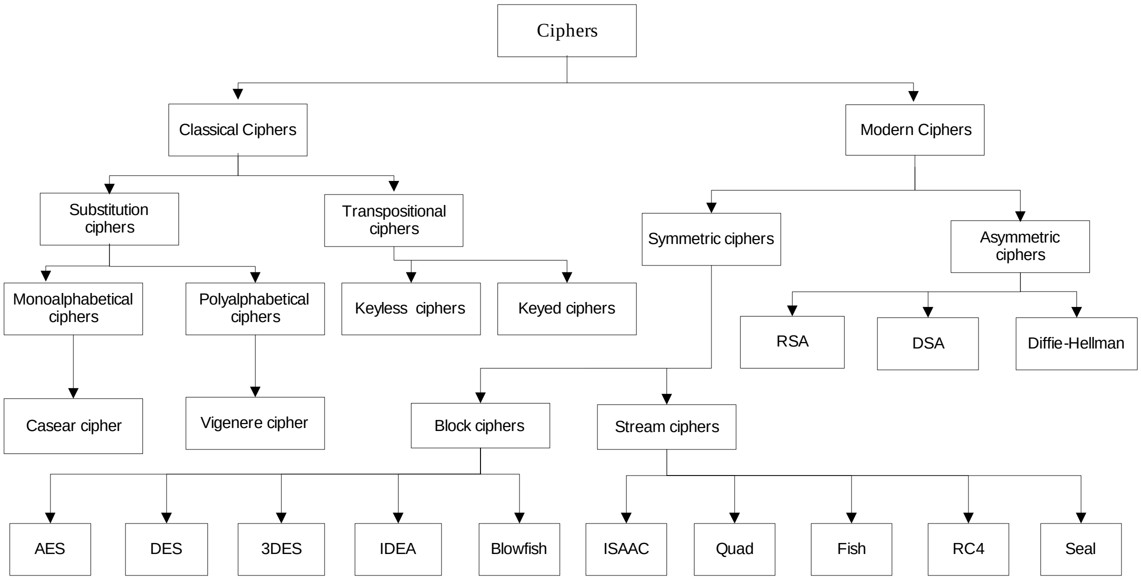


Fig. 3.3: Classifications of cryptographic algorithms.

3.3.3 Kerchoff's Principle

The security of the system has to be based on the assumption that the enemy has full knowledge of the design and implementation details of the steganographic system. The only missing information for the enemy is a short, easily exchangeable random number sequence, the secret key. Without this secret key, the enemy should not have the chance to even suspect that on an observed communication channel, hidden communication is taking place.

3.3.4 Diffusion and Confusion

Claude Shannon, in one of the fundamental papers on the theoretical foundations of cryptography [4, 5], gave two properties that a good cryptosystem should have to hinder statistical analysis: diffusion and confusion. Diffusion means that if we change a character of the plaintext, then several characters of the ciphertext should change, and similarly, if we change a character of the ciphertext, then several characters of the plaintext should change. This means that frequency statistics of letters in the plaintext are diffused over several characters in the ciphertext, which means that much more ciphertext is needed to do a meaningful statistical attack. Confusion means that the key does not relate in a simple way to the ciphertext. In particular, each character of the ciphertext should depend on several parts of the key.

3.3.5 Spatial and Frequency domain

Spatial Domain

In the spatial domain method, the pixel composing of image details are considered and the various procedures are directly applied on these pixels. The image processing functions in the spatial domain may be expressed as

$$g(x, y) = T[f(x, y)], \tag{3.7}$$

where $f(x, y)$ is the input image, $g(x, y)$ is the processed output image and T represents an operation on f defined over some neighborhood of (x, y). Sometimes T can also be used to operate on a set of input images. The spatial domain is the normal image space, in which a change in position in image I directly projects to a change in position in scene S. Distances in I (in pixels) correspond to real distances (e.g. in meters) in S. We can also discuss the frequency with which image values change, that is, over how many pixels does a cycle of periodically repeating intensity variations occur. One would refer to the number of pixels over which a pattern repeats (its periodicity) in the spatial domain.

Frequency Domain

The frequency domain is a space in which each image value at image position F represents the amount that the intensity values in image I vary over a specific distance related to F. In the frequency domain, changes in image position correspond to changes in the spatial frequency, (or the rate at which image intensity values) are changing in the spatial domain image I. In simple spatial domain, we directly deal with the image matrix, whereas in frequency domain, we deal an image like this.

Frequency Components Any image in spatial domain can be represented in a frequency domain. But what do these frequencies actually mean. We will divide frequency components into two major components.

High Frequency Components High frequency components correspond to edges in an image.

Low frequency Components Low frequency components in an image correspond to smooth regions.

Difference Between Spatial Domain and Frequency Domain In spatial domain, we deal with images as it is. The values of the pixels of image change with respect to scene, whereas in frequency domain, we deal with the rate at which the pixel values are changing in spatial domain.

3.3.6 Chaos and Cryptography

In a nonlinear definable systems the phenomena of chaos is seen that exhibit pseudo-random behavior and is highly sensitive to the initial conditions applied to the system. The stability of the system is an important parameter for user in applications with the definition of Lyapunov exponents. An important aspect of these systems is the understanding of the system output for an observer who is aware of the initial conditions governing the characteristics, and on the other hand, the system appears to be highly random if the preliminary inputs to the system are unknown. If the pseudo-random behavior is known to the legitimate owner of data, this characteristic can be used to substitute and diffuse plaintext in order to achieve resistance and protection against unauthorized entities. In addition to encryption of text, numerous formats of data are used in communication systems that need to be protected.

Fundamental Properties of Chaotic Systems

Chaos has been witnessed in many natural structures covering a significant amount of technical and industrial areas. These occurrences display definite possessions that mark them difficult and volatile. Chaos theory deals with constructions that progress in time to a specific kind of dynamical actions. Several authors have addressed the mathematical theory of chaos due to its vast and most applicable effects in various fields of science. In broad spectrum, these schemes follow a definite set of procedures of improvement. Generally, chaos happens simply in certain deterministic nonlinear systems. Clearly, chaos seems when there is a continuous and disorganized looking long term progression that fulfills definite mathematical benchmarks. There are certain set of properties that sum up the features witnessed in chaotic systems. These measured the mathematical principles that describe chaos. The most appropriate are [111]:

1. **Nonlinearity:** If a system is linear, it cannot be chaotic.
2. **Determinism:** It has deterministic fundamental rules that every future state of the system must follow.
3. **Sensitivity to initial conditions:** Slight deviations in its early state can lead to completely dissimilar performance in its last state.

4. **Continued irregularity in the actions of the system:** Secret order together with a large or infinite amount of unstable periodic designs. This unseen direction forms the structure of irregular chaotic systems.

5. **Long-term prediction:** It is commonly difficult due to sensitivity to initial conditions, which can be recognized only to a limited amount of accuracy.

Table 3.1: Comparison of chaotic and cryptographic properties.

Chaos theory	**Cryptography**
Chaotic system	Pseudo-chaotic system
Nonlinear transform	Nonlinear transform
Infinite number of states	Finite states
Initial state	Plaintext
Final state	Ciphertext
Initial conditions and/or parameter	Key
Asymptotic independence of initial and final states	Confusion
Sensitivity to initial conditions and parameters mixing	Diffusion

3.4 Fundamentals of Watermarking

In recent years, digital data is obtained and transmitted easily. This ease has instigated the wide appearance, transmission, and storage of digital data. The technologies that have supported this flooding of digital data are internet, World Wide Web (www), CD-ROM, and DVD. Although the widespread use of digital data has brought a lot of ease in different aspects, nonetheless, it is not without its side effects. These side effects are best presented by asking a question: with the digital data being so widely used, how are we going to address the issues like privacy, copyright infringement, authentication, and security? Three different technologies; information hiding, steganography and watermarking, are mostly used to address issues like these. These three technologies often use similar technical approaches and are closely related [72]. However, they do have some philosophical differences that affect their design towards a problem. Watermarking is defined as the practice of imperceptibly altering a work to embed a message about that work, whereas steganography represents the art of concealed communication.

Here, the very existence of a message being kept secret. On the other hand, data hiding is a more general term and encompasses a wide range of problems that are either related to making information imperceptible or secret. A detailed discussion about the differences in these concepts can be found in [73].

3.5 Digital Watermarking

Digital watermarking is a process of embedding unobtrusive marks or labels into digital content. These embedded marks are typically imperceptible (invisible) that can later be detected or extracted (Yeung, Yeo, & Holliman, 1998). The basic principle of the current watermarking systems are comparable to symmetric encryption as to the use of the same key for encoding and decoding of the watermark. Each watermarking system consists of two subsystems: a watermarking encoder and decoder (see Fig. 3.4). The formal definition of watermarking is given as follows:

Definition 91 *A watermarking system can be described by a pentraple* (C, W, K, E_K, D_K), *where* C *is the set of all original data,* W *the set of all watermarks,* K *is the set of all keys,* C' *is the set of all original data with watermark. The two functions* $E_K : C \times W \times K \rightarrow C^{'}$, $D_K : C' \times K \rightarrow W$, *describe the embedding and detecting process.*

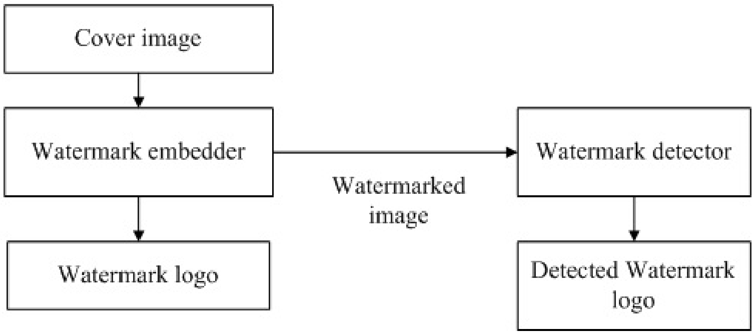


Fig. 3.4: A general procedure for watermarking scheme.

3.5.1 Basic Terminologies of Watermarking

The general definitions of some common terms used in the area of watermarking are listed below:

Watermark The information to be hidden. The term watermark also contains a hint that the hidden information is transparent like water.

Cover Media/Data The media used for carrying the watermark. Sometimes the terms original media, cover media and host media are also used to express it.

Watermark Data The digital medium which contains the watermark.

Extraction The procedure used for extracting the embedded watermark from the watermark object.

Detection The procedure used for detecting whether the given media containing a particular watermark.

3.5.2 Properties of Digital Watermarking

Effectiveness This is the probability that the message in a watermarked image will be correctly detected.

Fidelity Watermarking is a process that alters an original image to add a message to it, therefore it inevitably affects the image's quality.

Payload Size Every watermarked work is used to carry a message. The size of this message is often important as many systems require a relatively big payload to be embedded in a cover work.

Robustness There are many cases in which a watermarked work is altered during its lifetime, either by transmission over a lossy channel or several malicious attacks that try to remove the watermark or make it undetectable.

3.6 Classification of Watermarking

In this section, we discussed the classifications of watermarking with respect to different characteristics that are currently available for real time applications (see Fig. 3.5).

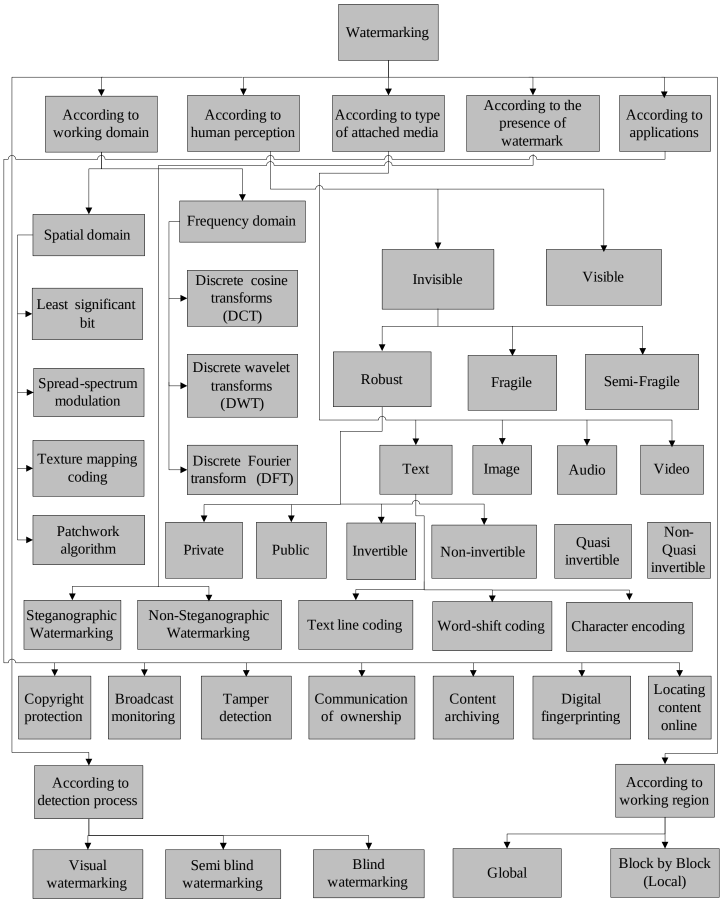


Fig. 3.5: Classifications of watermarking with respect to different aspects.

3.6.1 Classification Established on Human Perception

This is subdivided into visible watermarks and invisible watermarks.

Visible Watermarks

These watermarks can be seen clearly by the viewer and can also identify the logo or the owner. Visible watermarking technique changes the original signal. The watermarked signal is different from the original signal. Visible watermark embedding algorithms are less computationally complex. The watermarked image cannot with stand the signal processing attacks, like the watermark can be cropped from the watermarked image. Spreading the watermark throughout the image is a best option, but the quality of the image is degraded which prevents the image from being used in medical applications.

Invisible Watermarks

These watermarks cannot be perceived by the observer. The output signal does not change much when compared to the original signal. The watermarked signal is almost similar to the original signal. As the watermark is invisible, the imposter cannot crop the watermark as in visible watermarking. Invisible watermarking is more robust to signal processing attacks when compared to visible watermarking. As the quality of the image does not suffer much, it can be used in almost all the applications [74]. The invisible watermarking is further classified into fragile, semi-fragile and robust watermarks.

Fragile Watermarks These watermarks are very sensitive. They can be destroyed easily with slight modifications in the watermarked signal.

Semi-Fragile Watermarks These watermarks are broken if the modifications to the watermarked signal exceed a pre-defined user threshold. If the threshold is set to zero, then it operates as a fragile watermark. This method can be used to ensure data integrity and also data authentication [75].

Robust Watermarks These watermarks cannot be broken easily as they withstand many signal processing attacks. Robust watermark should remain intact permanently in the embedded signal such that attempts to remove or destroy the robust watermark will degrade or even may destroy the quality of the image. This method can be used to ensure copyright protection of the signal [75]. The robust watermarking is majorly sub-divided into public watermarks and private watermarks.

Public Watermarks In this watermarking, the user is authorized to detect the watermark embedded in the original signal.

Private Watermarks In this watermarking, the user is not authorized to detect the watermark embedded in the original signal.

3.6.2 Classification Established on Detection Process

To detect the embedded data [76] Based on the level of required information all watermarks are subdivided into blind watermarks, semi-blind watermarks and non-blind watermarks.

Blind Watermarks

These watermarks detect the embedded information without the use of original signal. They are less robust to any attacks on the signal.

Semi-Blind Watermarks

These watermarks require some special information to detect the embedded data in the watermarked signal.

Non-Blind Watermarks

These watermarks require the original signal to detect the embedded information in the watermarked signal. They are more robust to any attacks on the signal when compared to blind watermarks.

3.6.3 Classification Established on Information of Existence of the Watermark

The classification established on information of existence of the watermark is sub-divided into steganographic watermarking and non-steganographic watermarking.

Steganographic Watermarking

The user is not aware of the presence of the watermark.

Non-Steganographic Watermarking

The user is aware of the presence of the watermark.

3.6.4 Classification of Watermarking Techniques on the Basis of Working Domains

There are two major techniques for watermarking based on working domains:

Spatial Domain

This domain focuses on modifying the pixels of one or two randomly selected subsets of images. It directly loads the raw data into the image pixels. Some of its main algorithms are:

Least Significant Bit Old popular technique embeds the watermark in the LSB of pixels. This method is easy to implement and does not generate serious distortion to the image; however, it is not very robust against attacks. The embedding of the watermark is performed choosing a subset of image pixels and substituting the least significant bit of each of the chosen pixels with watermark bits. The watermark may be spread throughout the image or may be in the select locations of the image. But these primitive techniques are vulnerable to attacks and the watermark can be easily destroyed. Such an approach is very sensitive to noise and common signal processing and cannot be used in practical applications.

SSM Modulation Based Technique Spread-spectrum techniques are methods in which energy generated at one or more discrete frequencies is deliberately spread or distributed in time. SSM based watermarking algorithms embed information by linearly combining the host image with a small pseudo noise signal that is modulated by the embedded watermark.

Texture Mapping Coding Technique This method is useful in only those images which have some texture part in it. This method hides the watermark in the texture part of the image. This algorithm is only suitable for those areas with large number of arbitrary texture images (disadvantage) [77], and cannot be done automatically. This method hides data within the continuous random texture patterns of a picture.

Patchwork Algorithm Patchwork is a data hiding technique developed by Bender et. al. and published on IBM Systems Journal [75]. It is based on a pseudorandom, statistical model. Patchwork imperceptibly inserts a watermark with a particular statistic using a Gaussian distribution. A pseudo randomly selection of two patches is carried out where the first one is A and the second is B. Patch A image data is brightened where as that of patch B is darkened (for purposes of this illustration this is magnified).

Frequency Domain

This technique is also called transform domain. Values of certain frequencies are altered from their original. There are several common used transform domain methods, such as

Discrete Cosine Transforms (DCT) DCT based watermarking techniques are more robust compared to simple spatial domain watermarking techniques. Such algorithms are robust against simple image processing operations like low pass filtering, brightness and contrast adjustment, blurring etc. However, they are difficult to implement and are computationally more expensive. At the same time they are weak against geometric attacks like rotation, scaling, cropping etc.

Steps in DCT Block Based Watermarking Algorithm

i. Segment the image into non-overlapping blocks of 8 × 8,

ii. Apply forward DCT to each of these blocks,

iii. Apply some block selection criteria (e.g. HVS),

iv. Apply coefficient selection criteria (e.g. highest),

v. Embed watermark by modifying the selected coefficients,

vi. Apply inverse DCT transform on each block

Discrete Wavelet Transforms (DWT) Wavelet Transform is a modern technique frequently used in digital image processing, compression, watermarking etc. The transforms are based on small waves, called wavelet, of varying frequency and limited duration. The wavelet transform decomposes the image into three spatial directions, i.e. horizontal, vertical and diagonal. Hence wavelets reflect the anisotropic properties of HVS more precisely. Magnitude of DWT coefficients is larger in the lowest bands (LL) at each level of decomposition and is smaller for other bands (HH, LH, and HL).

Advantages of DWT over DCT Wavelet transform understands the HVS more closely than the DCT. Wavelet coded image is a multi-resolution description of image. Hence an image can be shown at different levels of resolution and can be sequentially processed from low resolution to high resolution.

Disadvantages of DWT over DCT Computational complexity of DWT is more compared to DCT. As Feig [78] pointed out it only takes 54 multiplications to compute DCT for a block of 8x8, unlike wavelet calculation depends upon the length of the filter used, which is at least 1 multiplication per coefficient.

Discrete Fourier Transform (DFT)

Transforms a continuous function into its frequency components. It has robustness against geometric attacks like rotation, scaling, cropping, translation etc. DFT shows translation invariance. Spatial shifts in the image affects the phase representation of the image but not the magnitude representation, or circular shifts in the spatial domain don't affect the magnitude of the Fourier transform.

Advantages of DFT over DWT and DCT DFT is rotation, scaling and translation (RST) invariant. Hence it can be used to recover from geometric distortions, whereas the spatial domain, DCT and the DWT are not RST invariant and hence it is difficult to overcome from geometric distortions.

3.7 Digital Watermarking Applications

3.7.1 Copyright Protection

Designed to prevent there production of software, films, music, and other media,usually for copyright reasons.

3.7.2 Broadcast Monitoring

With the global television and radio landscape changing more quickly than ever before, how can content owners effectively manage their media assets and ensure fair compensation?

3.7.3 Locating Content Online

It has also become a primary sales tool and selling environment, providing an opportunity to showcase our products or services and attract buyers from around the world.

3.7.4 Communication of Ownership and Copyright

In our cyber culture, digital has become a primary means of communication and expression. The combination of access and new tools enables digital content to travel faster and further than ever before as it is uploaded, dispersed, viewed, downloaded, modified and reproposed at breathtaking speed.

3.7.5 Content Archiving

Watermarking can be used to insert digital object identifier or serial number to help archive digital contents like images, audio or video.

3.7.6 Meta Data Insertion

Meta-data refers to the data that describes data. Images can be labeled with its content and can be used in search engines. Medical X-rays could store patient records.

3.7.7 Tamper Detection

Digital content can be detected for tampering by embedding fragile watermarks. If the fragile watermark is destroyed or degraded, it indicated the presence of tampering and hence the digital content cannot be trusted.

3.7.8 Digital Fingerprinting

Digital Fingerprinting is a technique used to detect the owner of the digital content. Fingerprints are unique to the owner of the digital content. Hence a single digital object can have different fingerprints because they belong to different users.

3.8 Steganography Fundamentals and Techniques

Steganography word is originated from Greek words Steganós (Covered), and Graptos (Writing) which literally means “cover writing” [79]. Generally steganography is known as “invisible” communication. Steganography means to conceal messages existence in another medium (audio, video, image, communication). Today’s steganography systems use multimedia objects like image, audio and video etc. as cover media because people often transmit digital images over email or share them through other internet communication application. It is different from protecting the actual content of a message. In simple words it would be like that, hiding information into other information. Steganography means is not to alter the structure of the secret message, but hides it inside a cover-object (carrier object). After hiding process cover object and stego-object (carrying hidden information object) are similar. So, steganography (hiding information) and cryptography (protecting information) are totally different from one another. Due to invisibility or hidden factor it is difficult to recover information without known procedure in steganography. Detecting procedure of steganography known as Steganalysis.

3.8.1 Steganography versus Cryptography

It is often thought that communications may be secured by encrypting the traffic, but this has rarely been adequate in practice [81]. Cryptography deals with the encryption of text to form cipher (encrypted) text using a secret key. However, the transmission of cipher text may easily arouse attackers suspicion, and the cipher text may thus be intercepted, attacked or decrypted violently. In order to overcome the shortcomings of cryptographic techniques, an important sub-discipline of information hiding i.e. Steganography has been developed as a new covert communication means in recent years. It transfers message secretly by embedding it into a cover medium with the use of information hiding techniques [82].

Steganography, hides the existence of message such that intruder can't even guess that communication is going on and thus provides a higher level of security than cryptography. Both cryptographic and steganographic systems provide secret communications, but they are different in terms of system breaking. If the intruder can read the secret message, then a cryptographic system is broken.

However, a steganographic system is considered broken if the intruder can detect the existence or read the contents of the hidden message. If the intruder suspects a specific file or steganography method even without decoding the message, a steganographic system will be considered to have failed. Thus, steganographic systems are more fragile than cryptography systems in terms of system failure [83].

3.8.2 Steganography versus Watermarking

Differences between steganography and watermarking are both subtle and essential.

The main goal of steganography is to hide a message m in some audio or video (cover) data d, to obtain new data d', practically indistinguishable from d, by people, in such a way that an eavesdropper cannot detect the presence of m in d'.

The main goal of watermarking is to hide a message m in some audio or video (cover) data d, to obtain new data d', practically indistinguishable from d, by people, in such a way that an eavesdropper cannot remove or replace m in d'. It is also often said that the goal of steganography is to hide a message in *one-to-one communications* and the goal of watermarking is to hide message in *one-to-many communications.*

Shortly, one can say that cryptography is about protecting the content of messages, steganography is about concealing its very existence. Steganography methods usually do not need to provide strong security against removing or modification of the hidden message. Watermarking methods need to be very robust to attempts to remove or modify a hidden message.

3.8.3 Basic Terminologies of Stegosystems

- **Covertext / Cover Media:** (cover-data - cover-object): The covertext is an original unaltered message. In other words, cover media is the medium in which message is embedded to hide the presence of secret data or used as the carrier to embed message into.
- **Stego / Stegotext (stego-data - stego-object):** The media through which the data is hidden Or generated data which is carrying a hidden message.
- **Secret data:** The data to be hidden or extract.
- **Embedding process**: A process in which the sender tries to hide a message by embedding it into a (randomly chosen) cover-text, usually using a key, to obtain a stego-text (stego-data or stego-object). The embedding process can be described by the mapping $E : C \times K \times M \rightarrow C$, where C is the set of possible cover- and stego-texts, K is the set of keys and M is the set of messages.
- **Recovering process (extraction process):** A process in which the receiver tries to get using the key only, not the covertext, the hidden message in the stegotext. The recovery process can be seen as mapping $D : C \times K \rightarrow C$.
- **Security requirement:** is that a third person watching such a communication should not be able to find out whether the sender has been active, and when, in the sense that he really embedded a message in the cover -text. In other words, stegotexts should be indistinguishable from covertexts.
- **Steganalysis:** The process by which secret data is to be extracted.

Definition 92 (Stegosystem) *Let $\mathcal{C}$ be a distribution on a set C of covertexts. A stegosystem is a triple of probabilistic polynomial-time algorithms (S_K, S_E, S_D) with the following properties:*

- *The key generation algorithm S_K takes as input the security parameter n and outputs a bit string s_k, called the stego-key.*
- *The steganographic encoding algorithm S_E takes as inputs the security parameter n, the stego key s_k and a message $m \in \{0,1\}^l$ to be embedded and outputs an element c of the covertext space C, which is called stegotext. The algorithm may access the covertext distribution $\mathcal{C}$.*
- *The steganographic decoding algorithm S_D takes as inputs the security parameter n, the stego key s_k, and an element c of the covertext space C and outputs either a message $m \in \{0,1\}^l$*

or a special symbol $\perp$. An output value of $\perp$ indicates a decoding error, for example, when S_D has determined that no message is embedded in c. For all s_k output by $S_K(1^n)$ and for all $m \in \{0,1\}^l$, the probability that $S_D(1^n, s_k, S_E(1^n, s_k, m)) = m$ must be negligible in n.

3.9 Types of Steganography

In the literature there are basically three types of steganographic protocols: pure steganography, secret key steganography, and public key steganography; the latter is based on principles of public key cryptography. In the following subsections, all three types will be discussed.

3.9.1 Pure Steganography

We call a steganographic system which does not require the prior exchange of some secret information (like a stego-key) pure steganography. Formally, the embedding process can be described as a mapping $E : C \times M \rightarrow C$, where C is the set of possible covers and M the set of possible messages.

The extraction process consists of a mapping $D : C \rightarrow M$, extracting the secret message out of a cover. Clearly, it is necessary that $|C| \geq |M|$. Both sender and receiver must have access to the embedding and extraction algorithm, but the algorithms should not be public.

Definition 93 *The quadruple $< C, M, D, E >$, where C is the set of possible covers, M the set of secret messages with $|C| \geq |M|$, $E : C \times M \rightarrow C$the embedding function and $D : C \rightarrow M$, the extraction function, with the property that $D(E(c,m)) = m$ for all $m \in M$ and $c \in C$ is called a pure steganographic system.*

In most practical steganographic systems the set C is chosen to consist of meaningful, and apparently harmless messages (like the set of all meaningful digital images), two communication partners would be able to exchange without raising suspicion. The embedding process is defined in a way that a cover and the corresponding stego-object are perceptually similar. Formally, perceptual similarity can be defined via a similarity function:

3.9.2 Secret Key Steganography

With pure steganography, no information (apart from the functions $E$and D) is required to start the communication process; the security of the system thus depends entirely on its secrecy. This is not very secure in practice because this violates Kerckhoffs' principle (see). So we must assume that Wendy knows the algorithm Alice and Bob use for information transfer. In theory, she is able to extract information out of every cover sent between Alice and Bob. The security of a steganographic system should thus rely on some secret information traded by Alice and Bob, the stego-key. Without knowledge of this key, nobody should be able to extract secret information out of the cover.

A secret key steganography system is similar to a symmetric cipher: the sender chooses a cover c and embeds the secret message into c using a secret key k. If the key used in the embedding process is known to the receiver, he can reverse the process and extract the secret message. Anyone who does not know the secret key should not be able to obtain evidence of the encoded information. Again, the cover c and the stego-object can be perceptually similar.

Definition 94 *The quintuple $< C, M, K, D_K, E_K >$, where C is the set of possible covers, M the set of secret messages with $|C| \geq |M|$, K the set of secret keys, $E_K : C \times M \times K \to C$ and $D_K : C \times K \to M$ with the property that $D_K(E_K(c, m, k), k) = m$ for all $m \in M$, $c \in C$ and $k \in K$, is called a secret key steganographic system.*

Secret key steganography requires the exchange of some key, although the transmission of additional secret information subverts the original intention of invisible communication. So as in cryptography, we assume that all communication parties are able to trade secret keys through a secure channel. Alice and Bob could agree on a stego-key before imprisonment. However, by using some characteristic features of the cover and a secure hash function H it is possible to calculate a key used for secret communication directly out of the cover: $k = H$ (feature). If the embedding process does not change the "feature," the receiver is able to recalculate the key. Obviously such a feature has to be highly "cover dependent" to reach an adequate level of security (however, the security depends on the secrecy of H, thus violating Kerckhoffs' principle again). If the cover is a digital image, one could take all most significant bits of the cover's color values as a "feature.” This method could be also used to calculate a secret session key out of a general key k' valid for a longer period of time, if the hash function depends on k'.

Some algorithms additionally require the knowledge of the original cover (or some other information not derivable from the stego-object) in the decoding phase. Such systems are of limited interest, because their use requires the transmission of the original cover, a problem strongly related to key-exchange in traditional cryptography. These algorithms can be seen as a special case of secret key steganographic systems in which $K = C$ or $K = C \times K'$ where K' denotes an additional set of secret keys.

3.9.3 Public Key Steganography

As in public key cryptography, public key steganography does not rely on the exchange of a secret key. Public key steganography systems require the use of two keys, one private and one public key; the public key is stored in a public database. Whereas the public key is used in the embedding process, the secret key is used to reconstruct the secret message.

One way to build a public key steganography system is the use of a public key cryptosystem. We will assume that Alice and Bob can exchange public keys of some public key cryptography algorithm before imprisonment (this is, however, a more reasonable assumption). Public key steganography utilizes the fact that the decoding function D in a steganography system can be applied to any cover c, whether or not it already contains a secret message (recall that D is a function on the entire set C). In the latter case, a random element of M will be the result; we will call it "natural randomness" of the cover. If one assumes that this natural randomness is statistically indistinguishable from ciphertext produced by some public key cryptosystem, a secure steganography system can be built by embedding ciphertext rather than unencrypted secret messages.

A protocol which allows public key steganography has been proposed by Anderson in [4, 5]; it relies on the fact that encrypted information is random enough to "hide in plain sight": Alice encrypts the information with Bob's public key to obtain a random-looking message and embeds it in a channel known to Bob (and hence also to Wendy), thereby replacing some of the "natural randomness" with which every communication process is accompanied. We will assume that both the cryptographic algorithms and the embedding functions are publicly known. Bob, who cannot decide a priori if secret information is transmitted in a specific cover, will suspect the arrival of a message and will simply try to extract and decrypt it using his private key. If the cover actually contained information, the decrypted information is Alice's message.

Since we assumed that Wendy knows the embedding method used, she can try to extract the secret message sent from Alice to Bob. However, if the encryption method produces random-looking ciphertext, Wendy will have no evidence that the extracted information is more than some random bits. She thus cannot decide if the extracted information is meaningful or just part of the natural randomness, unless she is able to break the cryptosystem.

3.10 Steganography in Digital Mediums

Depending on the type of the cover object there are many suitable steganographic techniques which are followed in order to obtain security.

3.10.1 Image Steganography

Taking the cover object as image in steganography is known as image steganography. Generally, in this technique pixel intensities are used to hide the information.

3.10.2 Network Steganography

When taking cover object as network protocol, such as TCP, UDP, ICMP, IP etc., where protocol is used as carrier, is known as network protocol steganography. In the OSI network layer model there exist covert channels where steganography can be achieved in unused header bits of TCP/IP fields [84].

3.10.3 Video Steganography

Video Steganography is a technique to hide any kind of files or information into digital video format. Video (combination of pictures) is used as carrier for hidden information. Generally discrete cosine transform (DCT) alter values (e.g., 8.667 to 9) which is used to hide the information in each of the images in the video, which is not noticeable by the human eye. Video steganography uses such as H.264, Mp4, MPEG, AVI or other video formats.

3.10.4 Audio Steganography

When taking audio as a carrier for information hiding it is called audio steganography. It has become very significant medium due to voice over IP (VOIP) popularity. Audio steganography uses digital audio formats such as WAVE, MIDI, AVI MPEG or etc. for steganography.

3.10.5 Text Steganography

General technique in text steganography, such as number of tabs, white spaces, capital letters, just like Morse code and etc. is used to achieve information hiding [84].

3.11 Image Steganography Terminologies

Image steganography terminologies are as follows:

- **Cover-Image:** Original image which is used as a carrier for hidden information.
- **Message:** Actual information which is used to hide into images. Message could be a plain text or some other image.
- **Stego-Image:** After embedding message into cover image is known as stego-image.
- **Stego-Key:** A key is used for embedding or extracting the messages from cover-images and stego-images.

Generally image steganography is method of information hiding into cover-image and generates a stego-image. This stego-image then sent to the other party by known medium, where the third party does not know that this stego-image has hidden message. After receiving stego-image hidden message can simply be extracted with or without stego-key (depending on embedding algorithm) by the receiving end [84]. Basic diagram of image steganography is shown in Figure 2 without stego-key, where embedding algorithm required a cover image with message for embedding procedure. Output of embedding algorithm is a stego-image which simply sent to extracting algorithm, where extracted algorithm unhidden the message from stego-image.

3.12 Image Steganography Classifications

Generally image steganography is categorized in following aspects [85]:

- **High Capacity:** Maximum size of information can be embedded into image.
- **Perceptual Transparency:** After hiding process into cover image, perceptual quality will be degraded into stego-image as compare to cover-image.
- **Robustness:** After embedding, data should stay intact if stego-image goes into some transformation such as cropping, scaling, filtering and addition of noise.
- **Temper Resistance:** It should be difficult to alter the message once it has been embedded into stego-image.
- **Computation Complexity:** How much expensive it is computationally for embedding and extracting a hidden message?

3.13 Image Steganographic Techniques

Image steganography techniques can be divided into following domains.

3.13.1 Spatial Domain Methods

There are many versions of spatial steganography, all directly change some bits in the image pixel values in hiding data. Least significant bit (LSB)-based steganography is one of the simplest techniques that hides a secret message in the LSBs of pixel values without introducing many perceptible distortions. Changes in the value of the LSB are imperceptible for human eyes. Spatial domain techniques are broadly classified into:

i. Least significant bit (LSB),

ii. Pixel value differencing (PVD),

iii. Edges based data embedding method (EBE),

iv. Random pixel embedding method (RPE),

v. Mapping pixel to hidden data method,

vi. Labeling or connectivity method,

vii. Pixel intensity based method,

viii. Texture based method,

ix. Histogram shifting methods,

General merits of spatial domain LSB technique are:

i. There is less chance for degradation of the original image,

ii. More information can be stored in an image,

Disadvantages of LSB technique are:

i. Less robust, the hidden data can be lost with image manipulation,

ii. Hidden data can be easily destroyed by simple attacks.

3.13.2 Transform Domain Technique

This is a more complex way of hiding information in an image. Various algorithms and transformations are used on the image to hide information in it. Transform domain embedding can be termed as a domain of embedding techniques for which a number of algorithms have been suggested [86]. The process of embedding data in the frequency domain of a signal is much stronger than embedding principles that operate in the time domain. Most of the strong steganographic systems today operate within the transform domain Transform domain techniques have an advantage over spatial domain techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing. Some transform domain techniques do not seem dependent on the image format and they may outrun lossless and lossy format conversions. Transform domain techniques are broadly classified into:

i. Discrete Fourier transformation technique (DFT),

ii. Discrete cosine transformation technique (DCT),

iii. Discrete Wavelet transformation technique (DWT),

iv. Lossless or reversible method (DCT),

v. Embedding in coefficient bits.

3.13.3 Distortion Techniques

Distortion techniques need knowledge of the original cover image during the decoding process where the decoder functions to check for differences between the original cover image and the distorted cover image in order to restore the secret message. The encoder adds a sequence of changes to the cover image. So, information is described as being stored by signal distortion [87]. Using this technique, a stego object is created by applying a sequence of modifications to the cover image. This sequence of modifications is use to match the secret message required to transmit [88]. The message is encoded at pseudo-randomly chosen pixels. If the stego-image is different from the cover image at the given message pixel, the message bit is a “1.” otherwise, the message bit is a “0.”

The encoder can modify the “1” value pixels in such a manner that the statistical properties of the image are not affected. However, the need for sending the cover image limits the benefits of this technique. In any steganographic technique, the cover image should never be used more than once. If an attacker tampers with the stego-image by cropping, scaling or rotating, the receiver can easily detect it. In some cases, if the message is encoded with error correcting information, the change can even be reversed and the original message can be recovered [89].

3.13.4 Masking and Filtering

These techniques hide information by marking an image, in the same way as to paper watermarks. These techniques embed the information in the more significant areas than just hiding it into the noise level. The hidden message is more integral to the cover image. Watermarking techniques can be applied without the fear of image destruction due to lossy compression as they are more integrated into the image.

Advantages of Masking and filtering Techniques:

i. This method is much more robust than LSB replacement with respect to compression since the information is hidden in the visible parts of the image.

Disadvantages of Masking and filtering Techniques:

ii. Techniques can be applied only to gray scale images and restricted to 24 bits.

3.14 Conclusion

The chief aim of this part of the thesis is to give a survey of existing information concealing methods, their focal points and hindrances. A few strategies for concealing information in text, image, and audio are portrayed, with appropriate introductions to the environment of each medium, as well as the qualities and shortcomings of each technique. Most information concealing frameworks exploit human perceptual shortcomings, yet have shortcomings they could call their own. In zones where cryptography and encryption are being prohibited, natives are taking a gander at steganography to go around such approaches and pass messages secretly. Business uses of steganography as computerized watermarks are at present being utilized to track the copyright and ownership of electronic media. This chapter additionally tells why information covering up is picking up significance nowadays and the objectives that must be accomplished by any information concealing procedure. The basics of cryptography, watermarking and steganography introduced in this chapter will help us equally in other parts of this dissertation .

Chapter 4

A Novel Technique for the Construction of Strong S-boxes Based on Chaotic Lorenz Systems

In cryptographic systems the encryption process relies on the nonlinear mapping of original data or plaintext to the secure data. The mapping of data is facilitated by the application of substitution process embedded in the cipher. It is desirable to have resistance against differential cryptanalysis, which assists in providing clues about the composition of keys, and linear cryptanalysis, where a simple approximation is created to emulate the original cipher characteristics. In this work, we propose the use of nonlinear chaos-based substitution boxes which employs continuous time Lorenz system and linear fractional transformation. The proposed substitution system eliminates the need of independent round keys in a substitution-permutation network. The performance of the new substitution box is evaluated by nonlinearity analysis, strict avalanche criterion, bit independence criterion, linear approximation probability and differential approximation probability.

4.1 Chaotic Lorenz System

The Lorenz system is inspired by the model or air flow in atmosphere in 1950 [96] and is the first numerical study on chaos. The system dynamics are represented by the following equation:

$$\begin{aligned} \frac{dx}{dt} &= \alpha(y-x), \\ \frac{dy}{dt} &= \beta x - y - xz, \\ \frac{dz}{dt} &= xy - \gamma z. \end{aligned} \tag{4.1}$$

The space plots resulting from Eq. 4.1 are shown in Figs. $4.1 - 4.4$. The values of the parameters are $\alpha = 10, \beta = 28$ and $\gamma = 8/3$. The intervals used for the states of the system are $-40 \leq x \leq 40,\ -40 \leq y \leq 40$, and $-40 \leq z \leq 40$. The system exhibits chaotic behavior for the selected parameters and intervals.

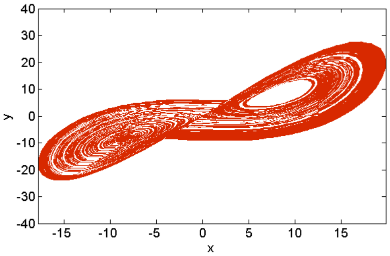


Fig. 4.1: The plot of Lorenz system along xy-axes for $\alpha = 10,\ \beta = 28,\ \gamma = 8/3$.

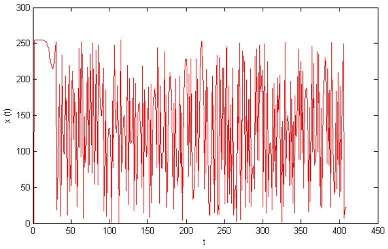


Fig. 4.2: The plot of Lorenz systems for x along t-axis for $\alpha = 10,\ \beta = 28,\ \gamma = 8/3$.

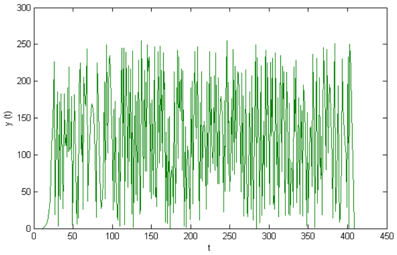


Fig. 4.3: The plot of Lorenz systems for y along t-axis for $\alpha = 10,\ \beta = 28,\ \gamma = 8/3$.

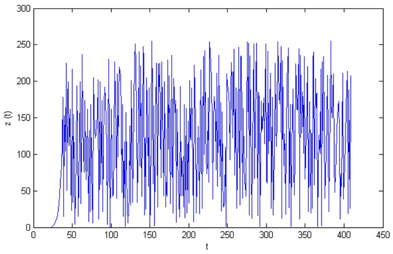


Fig. 4.4: The plot of Lorenz systems for z along t-axis for $\alpha = 10,\ \beta = 28,\ \gamma = 8/3$.

4.1.1 Chaos Based Algorithm for S-Box Design

The algorithm of the chaos based S-box design is presented in Fig. 4.5. The algorithm is divided into two parts: diffusion and substitution. In the proposed algorithm, the first two steps describe the diffusion process, whereas, the remaining portion depicts the realization of S-box.

Algorithm

A. 1: *System trajectories are obtained by solving the Lorenz system with selected initial conditions and chaotic parameter values employing four-step Runge–Kutta method.*

A. 2: *Selected trajectory is sampled at every (number of data/256) step.*

A. 3: *Use the linear fractional transformation [106], outputs corresponding to each sample is coded starting from 0 to 255.*

A. 4: *S-Box is generated using the codes corresponding to outputs with the code corresponding to the smallest output being the first cell of the S-Box.*

A. 5: *After the S-Box is generated, the rows are shifted to the left except for the first row. Cells of the remaining rows are shifted to the left such that at each row number of cells being shifted to the left incremented by one compared to the previous row.*

A. 6: *Once the rows are shifted, column wise rotation is performed starting from the last column leaving it un-rotated. Cells of the remaining columns are rotated such that at each column the number of cells being rotated is incremented by one compared to the previous column.*

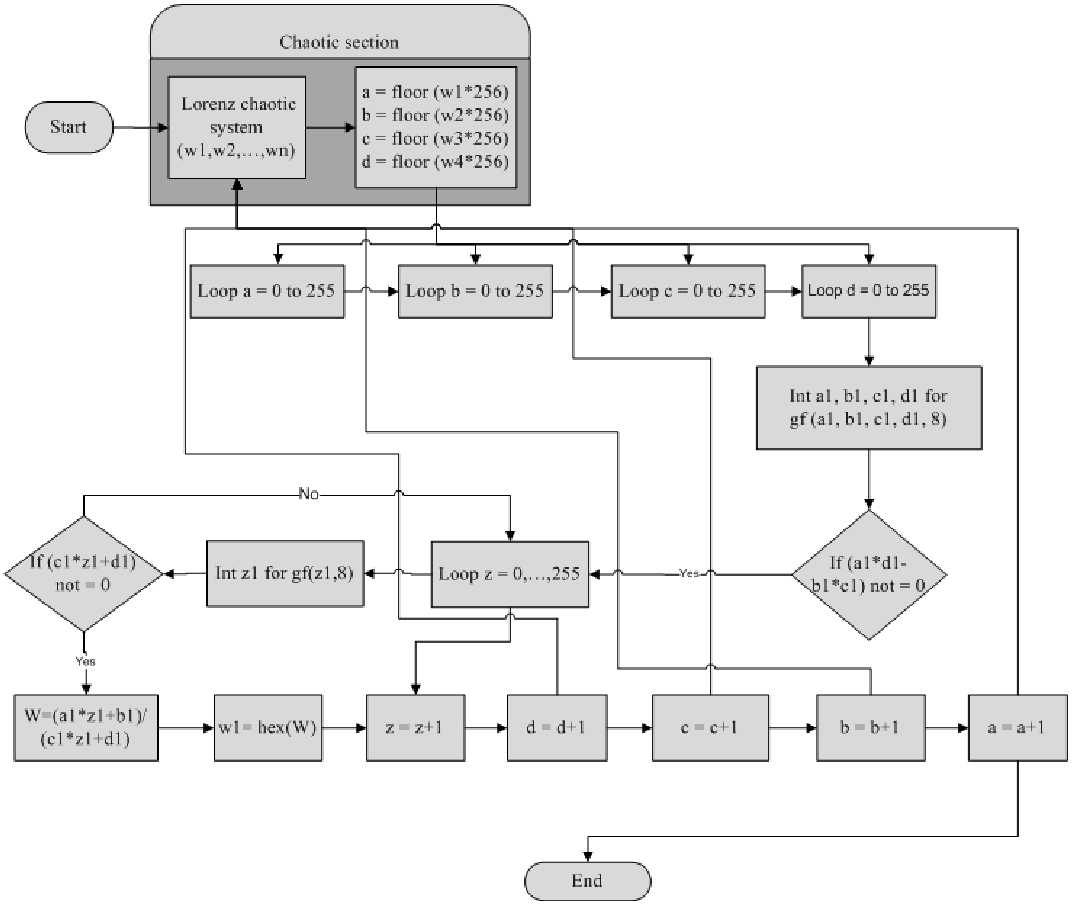


Fig. 4.5: Flow chart of proposed chaotic S-box.

In the diffusion process, the system trajectories are evaluated by the solution of Lorenz chaotic system. The number of orbits obtained depends on the dimension of the system, and is selected as a design parameter. The initial conditions of the system are selected at this stage. The Runge-Kutta method is applied to generate the chaotic parameters. A trajectory is selected and sampled at 8 bit resolution. The objective is to construct an S-box capable of substituting 8 bits of data, as a result, 256 samples are generated. Thus, coded samples, used in the S-box, range from 0 to 255. The entries of the S-box are populated by using the codes generated the samples generated by the selected system trajectory. A coding table is used to assign the corresponding entry into the S-box by selecting minimum output value in comparison to the samples utilized in the first cell. For example, in order to construct an S-box of dimension 4 × 4, sixteen samples are generated from a selected orbit. The data from the selected orbit is shown in Figs. 4.1 − 4.3. A coding table is used to map the sampled values from the output of the Lorenz system to an entry in S-box (see Table 4.1).

In this work, the system trajectory is generated for 1000 data samples while keeping the values of initial conditions as $x = 1,\ y = 0$, and $z = 0$. In order to ignore the transients of the chaotic system, first 1000 samples are ignored. The resulting S-box based on chaotic system is presented in Table 4.1.

Table 4.1: Algebraic structure of S-box in the form of 16 × 16 matrix.

3	167	29	139	249	80	34	165	250	251	238	110	33	38	140	17
0	41	135	164	236	71	16	209	99	143	151	70	188	184	252	242
60	120	231	105	49	66	128	121	125	218	178	196	89	154	244	192
155	82	162	185	138	97	213	50	10	113	54	237	183	22	202	194
208	191	129	136	197	137	26	152	168	103	13	65	132	39	79	61
119	160	44	207	102	175	95	72	74	235	55	63	247	144	203	20
8	177	223	92	254	90	228	118	224	219	117	240	7	6	19	147
21	186	241	48	1	216	122	93	69	73	5	15	158	114	106	187
88	130	87	68	78	98	245	47	84	234	176	141	255	51	149	53
225	214	123	35	28	166	233	220	248	211	101	45	198	115	77	52
94	193	86	133	76	85	67	200	226	14	62	4	40	146	239	126
36	230	148	150	11	75	56	153	96	215	30	145	25	100	58	174
181	172	190	57	163	64	171	124	217	111	18	131	31	243	195	253
246	182	201	104	221	27	109	107	232	157	199	83	161	42	227	112
179	159	12	210	169	127	170	189	2	206	108	204	173	23	81	116
229	91	24	37	32	43	134	222	59	142	180	205	9	46	156	212

4.2 Performance Analysis of Chaotic S-Box

It is vital to assess the performance of the proposed S-box in an effort to establish its usefulness in encryption. Several properties are listed in literature, which indicate the strength of any S-box [101]. Among some of the prevailing methods used by cryptanalysis include differential analysis used for the analysis of DES [67] and information theoretic analysis with excerpts from the original concepts presented by Shannon [4]. In this work, we analyze the proposed S-box for five different properties, which includes nonlinearity, strict avalanche criterion (SAC), bit independence criterion (BIC), linear approximation probability and differential approximation probability. In order to determine the strength of the proposed S-box, the results of these analyses are prudently analyzed. In the following subsections, we present the details of these analyses and discuss the results pertaining to the strength the S-box under analysis.

4.2.1 Nonlinearity

In the nonlinearity analysis, the constituent Boolean functions are assessed with reference to the behavior of the input/output bit patterns. The set of all affine functions is used to compare the distance from the given Boolean function. Once the initial distance is determined, the bits in the truth table of the Boolean function are modified to approximate to the closest affine function. Number of modifications required to reach the closest affine functions bears useful characteristics in determining the nonlinearity of the transformation used in encryption process. The measure of nonlinearity is bounded by [113],

$$N_g = 2^{m-1}\left(1 - 2^m \max|S_g(w)|\right). \tag{4.2}$$

The Walsh spectrum $S_g(w)$ is defined as

$$S_g(w) = \sum_{w \in \mathbf{F}_{2^m}} (-1)^{g(x) \otimes \chi \cdot w}. \tag{4.3}$$

The S-boxes are defined in Galois filed F_{2^m} and the most favorable value that N has is 120.

Table 4.2: The results of nonlinearity analysis of S-boxes.

S-boxes	Nonlinearities
Proposed S-box	105.25
Wang[97]	104
Chen[95]	100
Tang[99]	100
Jakimoski[100]	98

4.2.2 Strict Avalanche Criterion Analytically

In Strict Avalanche Criterion (SAC), the behavior of the output bits is analyzed that results from a change at the input bit applied to the nonlinear S-box transformation. It is desired that almost half of the output bits change their value or simply toggle their state in response to a single change at the input. The change in the output bit patterns cause a series of variations in the entire substitution-permutation network (S-P network) and thus causes an avalanche effect. The extent of these changes assists in determining the resistance to cryptanalysis and the strength of the cipher.

Table 4.3: The results of Strict avalanche criterion for proposed S-box.

0.5156	0.4687	0.4843	0.4375	0.5468	0.5000	0.4531	0.4375
0.5468	0.5625	0.4843	0.4687	0.5156	0.5625	0.4687	0.5312
0.5156	0.4687	0.4687	0.5625	0.4062	0.5156	0.5000	0.4687
0.5156	0.5312	0.4843	0.4531	0.5156	0.5937	0.5000	0.5625
0.5781	0.5000	0.4687	0.4843	0.4375	0.4531	0.3806	0.5781
0.5156	0.5312	0.6093	0.5625	0.5312	0.4375	0.5312	0.5000
0.5468	0.5312	0.5468	0.5312	0.5312	0.6250	0.4375	0.4218
0.4531	0.4062	0.4843	0.5312	0.5156	0.5468	0.4843	0.5000

Table 4.4: Comparison of SAC analysis of proposed chaotic S-boxes with other S-Boxes.

S-boxes	SAC
Proposed S-box	0.4930
Wang[97]	0.4850
Chen[95]	0.4999
Tang[99]	0.4993
Jakimoski[100]	0.4972

4.2.3 Bit Independent Criterion

The Bit Independence Criterion (BIC) also relies on the changes at the input bits and the properties exhibited by the independence behavior of pair-wise input/output variables of avalanche vectors [120]. This criterion is analyzed by modifying single input bit from the plaintext.

Table 4.5: The Nonlinearity of BIC of proposed S-box.

....	102	106	102	94	92	96	96
102	....	106	106	104	102	96	100
106	106	....	102	104	106	106	104
102	106	102	....	102	102	102	100
94	104	104	102	....	96	100	94
92	102	106	102	96	....	98	96
96	96	106	102	100	98	....	96
96	100	104	100	94	96	96	....

Table 4.6: The dependent matrix in BIC of the proposed S-box.

....	0.4765	0.5273	0.5175	0.4843	0.5117	0.5097	0.4882
0.4765	....	0.5039	0.4785	0.5078	0.4960	0.5078	0.5312
0.5273	0.5039	....	0.4960	0.4912	0.4824	0.5097	0.4862
0.5175	0.4785	0.4960	....	0.4902	0.4862	0.5078	0.5097
0.4843	0.5078	0.4921	0.4902	....	0.5117	0.4960	0.5253
0.5117	0.4960	0.4824	0.4863	0.5117	....	0.5136	0.5000
0.5097	0.5078	0.5097	0.5078	0.4960	0.5136	....	0.4804
0.4882	0.5312	0.4863	0.5097	0.5253	0.5000	0.4804	....

Table 4.7. BIC of SAC analysis of S-boxes.

S-boxes	Average Values
Proposed S-boxes	0.476
AES	0.504
APA	0.499
Gray	0.502
Prime	0.502

4.2.4 Linear Approximation Probability

The imbalance of an event between input and output bits is quantified by the Linear approximation probability test [120]. In this method, the parity of the input bits given by a certain mask Γ_x and the parity of the output bits Γ_y are used to determine the probability of bias, and is given as,

$$L_p = \max_{\Gamma x, \Gamma y \neq 0} \left| \frac{\#\{x/x \cdot \Gamma_x = S(x) \cdot \Gamma_y\}}{2^n} - \frac{1}{2} \right|, \tag{4.4}$$

where Γ_x and Γ_y are the input/output masks used in determining the linear approximation probability. The total number of elements is given by 2^n and X is the set of all possible input.

Table 4.8: Linear approximation analysis of S-boxes.

LPA	S-boxes				
	Proposed	AES	APA	S_8 AES	Skipjack
Max L_p	0.140	0.062	0.062	0.062	0.109
Max Value	160	144	144	144	156

4.2.5 Differential Approximation Probability

It is desirable that the nonlinear transformation exhibit differential uniformity. In order to ensure the uniform mapping, a differential at the input, given as , uniquely maps to an output differential for all i. The differential approximation probability is mathematically defined as,

$$D_{p(\Delta x \to \Delta y)} = \left[\frac{\#\{x \in X/\ S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^n} \right]. \tag{4.5}$$

The proposed chaotic S-box is evaluated with differential approximation probability test. The results show that the performance of the new chaotic S-box is comparable to some of the commonly used

S-boxes.

Table 4.9: The differential approximation probability of proposed chaotic S-box ($D_1 - D_4$).

	0	1	2	3	4	5	6	7
0	0.031	0.031	0.031	0.023	0.031	0.023	0.031	0.046
1	0.031	0.031	0.031	0.031	0.031	0.039	0.031	0.023
2	0.023	0.031	0.031	0.031	0.031	0.031	0.031	0.023
3	0.031	0.039	0.031	0.039	0.031	0.039	0.031	0.031
4	0.031	0.031	0.031	0.031	0.039	0.500	0.031	0.046
5	0.031	0.031	0.031	0.031	0.023	0.031	0.031	0.031
6	0.031	0.031	0.031	0.031	0.031	0.023	0.031	0.031
7	0.031	0.031	0.031	0.039	0.031	0.031	0.031	0.031

D_1

	8	9	10	11	12	13	14	15
0	0.046	0.031	0.031	0.031	0.031	0.046	0.031	0.031
1	0.046	0.023	0.031	0.031	0.031	0.031	0.031	0.039
2	0.031	0.031	0.023	0.031	0.031	0.031	0.031	0.023
3	0.031	0.031	0.023	0.039	0.031	0.031	0.039	0.031
4	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031
5	0.031	0.031	0.031	0.031	0.039	0.023	0.031	0.023
6	0.031	0.023	0.031	0.031	0.023	0.023	0.039	0.031
7	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.039

D_2

	0	1	2	3	4	5	6	7
8	0.031	0.031	0.031	0.023	0.031	0.031	0.031	0.031
9	0.031	0.031	0.031	0.031	0.031	0.023	0.015	0.023
10	0.031	0.031	0.031	0.023	0.031	0.031	0.023	0.039
11	0.031	0.039	0.031	0.031	0.039	0.023	0.031	0.031
12	0.023	0.023	0.031	0.031	0.031	0.031	0.031	0.031
13	0.023	0.023	0.031	0.031	0.031	0.023	0.031	0.015
14	0.023	0.031	0.031	0.031	0.023	0.015	0.031	0.031
15	0.046	0.031	0.031	0.031	0.023	0.039	0.031	0.031

D_3

	8	9	10	11	12	13	14	15
8	0.031	0.023	0.023	0.031	0.031	0.046	0.031	0.031
9	0.031	0.023	0.023	0.031	0.031	0.046	0.031	0.031
10	0.031	0.023	0.031	0.031	0.023	0.031	0.023	0.039
11	0.031	0.039	0.031	0.023	0.023	0.031	0.031	0.023
12	0.046	0.039	0.031	0.031	0.031	0.031	0.031	0.031
13	0.031	0.023	0.031	0.023	0.023	0.023	0.023	0.023
14	0.031	0.023	0.031	0.023	0.031	0.023	0.023	0.031
15	0.023	0.023	0.031	0.039	0.031	0.031	0.031	....

D_4

Table 4.10: Comparison of differential approximation probability of proposed chaotic S-box with existing S-boxes.

	S-boxes				
	Proposed	AES	Gray	Skipjack	Xyi
Max. D_p	0.03	0.0156	0.0156	0.0468	0.0468

4.3 Results and Discussions

The comparison of the strong encryption capabilities show that the performance of the proposed S-box is comparable or superior to some prevailing S-boxes used in the area of cryptography. The nonlinearity analysis depicts that the properties are comparable to the S-boxes use as a benchmark in this work. The Table 4.2 presents a list of results of nonlinearity analysis. The result of SAC is very close 0.5, which assures the acceptability of this S-box to encryption applications. The results are shown in Table 4.4. In Table 4.7, a comparison of BIC is presented between the proposed S-box and AES, APA, Gray, Prime S-boxes. The results are in agreement with the desired range. In further analysis, the linear approximation analysis show that the new S-box conforms to the range of values specified for the good nonlinear components used for encryption applications. The results are shown in Table 4.8. Finally, the differential approximation probability analysis is presented in Table 4.9 and comparison with already existing S-boxes are in Table 4.10. In these tests, it is observed that the performance of the chaotic S-box is comparable to the existing well known S-boxes used as benchmarks in this chapter.

4.4 Conclusion

In this chapter, we present a method to construct new S-boxes with the application of Lorenz system along with linear fractional transformation. In order to evaluate the performance of the proposed S-box, a comparison is presented by the application of strict avalanche criterion, linear approximation probability, differential approximation probability, bit independent criterion and nonlinearity analysis. The existing S-boxes, which are used for the purpose of benchmarking, include AES, APA, Gray, Prime S-boxes. The results yield that the new S-box has desirable properties suitable for secure communications.

Chapter 5

Construction of S-box Based on Chaotic Boolean Function and its Application in Image Encryption

In numerous encryption frameworks, the first information is changed into encoded form by applying nonlinear substitutions and affecting diffusion. The goal of the nonlinear change is to accomplish high level of randomness in the image content. The choice of the source of randomness is critical because the success in cryptanalysis is demarked by the characteristics identified in the encrypted data. The chaotic frameworks show random conduct that is suitable for encryption applications where nonlinear transformations are needed in the middle of plaintext and the scrambled information. The application of nonlinear functional chaos based system with embedded chaotic systems and binary chaotic sequences can prompt randomness and diffusion in the information. In addition to the high state of randomness, the requirement for various round keys are needed in a run of the mill substitution-permutation process. The proposed strategy kills the requirement for different round keys, which is suitable for high speed communication frameworks. The measurable analyses performed on the proposed nonlinear algorithm which show improvement in encryption quality and safety against numerous brute-force and statistical attacks. Also, the proposed framework demonstrates high safety against differential and linear cryptanalysis.

5.1 Proposed Algorithm for Generating Chaotic S-boxes

The system proposed is as takes after. An 8-bit sequence of binary random variables is produced and is transformed into a decimal number; if the number exists then we will repeat the chaotic grouping progressively. Along these lines, a integer table in the range of $0 - 2^8$ can be obtained (see table 5.1). The algebraic expression of the proposed design is in the accompanying. Boolean function is a function that returns values 0 or 1. By and large, one takes a grouping of bits as information and produces 1 bit as a yield. The methodology is as takes after. The simplest mathematical objects that can display chaotic behavior are a class of one-dimensional maps:

$$\omega_{k+1} = \delta(\omega_k), \tag{5.1}$$

where $\omega_k = \delta^k(\omega_0) \in I,\ k = 0, 1, 2, ...$and $\delta^k : I \to I$ is nonlinear map. One parameter families of chaotic maps of the interval $[0, 1]$ with an invariant measure can be defined as the ratio of polynomials of degree N

$$\omega_N(x, \alpha) = \frac{\alpha^2 F}{1 + (\alpha^2 - 1)F}, \tag{5.2}$$

where α is the control parameter, F substitutes with the Chebyshev polynomial of the first kind and is the degree of Chebyshev polynomials. Hence,

$$\omega_N(x, \alpha) = \frac{\alpha^2 (T_N(\sqrt{x}))^2}{1 + (\alpha^2 - 1)(T_N(\sqrt{x}))^2}. \tag{5.3}$$

We used its conjugate or isomorphic map. Conjugacy means that the invertible map $h(x) = (x - 1)/x$ maps $I = [0, 1]$ into $[0, \infty)$. Using the hierarchy of families of one-parameter chaotic maps, we can generate new hierarchy of tripled maps with an invariant measure. In this chapter, one of the hierarchies of the chaotic map in the interval $[0, \infty)$ is adapted for constructing chaos-based hash function. Hence, this chaotic map can be defined as [119] :

$$\omega_N(x, \alpha_1) = \frac{1}{\alpha_1^2} \tan\left(N \arctan \sqrt{x_{k-1}}\right), \tag{5.4}$$

where α_1 is control parameter of the chaotic maps and N is the degrees of the Chebyshev polynomials. The mapping equation is

$$\omega = 0.b_1(\omega).b_2(\omega).b_3(\omega)...b_i(\omega)..., \tag{5.5}$$

where $\omega \in [0, 1],\ b_i(\omega) \in [0, 1]$. The ith bit $b_i(\omega)$ can be expressed as follows:

$$b_i(\omega) = \sum_{j=1}^{2^i - 1} (-1)^{j-1} \theta_\tau(\omega), \tag{5.6}$$

where $\theta_\tau(\omega)$ is a thresholding function can be defined as

$$\theta_\tau(\omega) = \begin{cases} 1, & \omega < \tau \\ 0, & \omega \geq \tau \end{cases}, \tag{5.7}$$

and its complementary function can be written as follows:

$$\overline{\theta}_{\tau}(\omega) = 1 - \theta_{\tau}(\omega). \tag{5.8}$$

Thus we can obtain a binary sequence

$$\begin{aligned} B_i^k &= l(\omega) = \left\{ b_i \left(\omega_N(x,\alpha) = \frac{\alpha^2 F}{1 + (\alpha^2 - 1)F} \right) \right\}, & (5.9) \\ &= \left\{ b_i \left(\omega_N(x,\alpha) = \frac{\alpha^2 (T_N(\sqrt{x}))^2}{1 + (\alpha^2 - 1)(T_N(\sqrt{x}))^2} \right) \right\}, & (5.10) \\ &= \left\{ b_i \left(\omega_N(x_k,\alpha_1) = \frac{1}{\alpha_1^2} \tan\left(N \arctan \sqrt{x_{k-1}} \right), \right) \right\}_{k=0}^{\infty}, & (5.11) \end{aligned}$$

where

$$x_k \in (0,1), \quad k = 0,1,2,... \tag{5.12}$$

Thus we can obtain a binary sequence which we call a chaotic bit sequence.

Table 5.1: The proposed chaotic S-box.

116	3	147	22	55	139	91	225	162	148	242	20	119	42	43	249
125	12	156	28	50	124	37	194	168	159	212	26	127	11	247	82
133	21	165	34	45	110	19	185	174	170	210	52	143	6	114	102
142	30	173	44	32	95	1	176	180	175	208	58	144	35	105	220
151	39	182	49	25	79	239	167	193	189	201	72	137	51	88	157
160	48	190	53	17	63	221	140	207	202	192	86	135	100	80	126
169	57	198	61	0	47	204	134	228	209	183	93	98	150	15	69
178	67	206	65	246	31	186	129	243	213	181	101	92	251	81	38
187	76	222	68	236	14	136	118	24	216	177	145	78	13	107	122
196	85	230	70	226	253	120	115	46	219	172	184	71	23	149	255
205	94	237	73	215	235	104	113	54	227	171	4	254	41	245	99
214	103	244	74	203	218	89	111	83	229	163	33	211	154	155	131
223	112	252	77	191	200	59	109	90	231	161	40	188	87	217	36
232	121	2	75	179	164	18	108	97	233	195	66	152	56	234	29
241	130	9	64	166	146	5	132	117	238	199	84	27	10	224	158
250	138	16	60	153	128	248	141	123	240	8	106	62	96	197	7

5.2 Some Cryptographic Properties of Boolean Functions

The aim of this section is to present a detailed and compact overview on the most essential aspects of Boolean functions and S-boxes related to cryptography [121, 122, 123].

Definition 95 *Let $V_2^n = GF(2)^n$ be the n-dimesional vector space over the binary field $GF(2)$. We call any function $g(x)$ from $GF(2)^n$ to $GF(2)$ to be the n variable Boolean function, where $x \in GF(2)^n, g(x) \in [0,1]$. An (n,m) function is a mapping $g = (g_0, g_1, g_2, ..., g_m)$ from $GF(2)^n$ to $GF(2)^m$ and every coordinates function g_i of an n variable boolean function. An (n,m) function is used in cryptosystem is often called an (n,m) S-box. In other words, an $n \times m$ S-box S is a mapping from $\{0,1\}^n$ to $\{0,1\}^m$ and can be represented by 2^n m bits number.*

5.2.1 Hamming Weight

Let g be n variables Boolean function. We call the numbers of $1's$ in a binary sequence, its Hamming weight that is denoted by $\mathbf{wt}(g)$ and defined by the following formula:

$$\mathbf{wt}(g) = \sum_{x=0}^{2^n-1} g(x). \tag{5.13}$$

5.2.2 Hamming Distance

Let g and h are $n-$variables Boolean functions. Then the Hamming distance between these functions can be calculated as follows:

$$d(g,h) = \sum_{x=0}^{2^n-1} g(x) \oplus h(x). \tag{5.14}$$

5.2.3 Algebraic Normal Form

The algebraic normal form (ANF) is an n-variables Boolean function $g(x)$ which can be written as follows:

$$g(x) = b \oplus b_0 x_0 \oplus b_{01} x_0 x_1 \oplus ... \oplus b_{012...n-1} x_0 x_1 x_2 ... x_{n-1}. \tag{5.15}$$

5.2.4 Algebraic Degree

The algebraic degree of a Boolean function $g(x)$, denoted by $deg(g(x))$, is defined to be the number of variables in the largest product term of the function's ANF having a non-zero coefficient.

5.2.5 Walsh Hadamard Transform

The Walsh Hadamard transform (WHT) of the truth table of a Boolean function $g(x)$, denoted by $\Omega(x)$, is a measure of the correlation between a function and the set of all linear functions. It is defined by

$$\Omega(x) = \sum_{x=0}^{2^n-1} (-1)^{g(x)\oplus l_\omega(x)}, \tag{5.16}$$

where Boolean function of the form $l_\omega(x) = \omega.x = \omega_0 x_0 \oplus \omega_1 x_1 \oplus ... \oplus \omega_{n-1}x_{n-1}$ is a linear function of n variables.

5.2.6 Balanced Boolean Function

A sequence $(0, 1)$ is called balanced if it contains an equal number of zeros and ones (one and minus one). A function is balanced if their sequence is balanced i.e. $wt(g) = 2^{n-1}$. In other words, a Boolean function is balanced if its output is equally distributed, i.e., its weight is equal to 2^{n-1}. This translates in $\Omega_g(\overline{0}) = 0$ for the Walsh spectrum.

5.2.7 Propagation Criteria

Let $g(x)$ be a Boolean function on $GF(2)^n$. If for $\alpha \in GF(2)^n$ function $g(x) \oplus g(x \oplus \alpha)$ is balanced, then the function $g(x)$ is said to have a propagation criteria with respect to the vector α. If $g(x)$ have a propagation criteria with respect to the all vectors with $0 < \mathbf{wt}(\alpha) \leq k$, then $g(x)$ has propagation criteria of degree k denoted by $PC(k)$. If $g(x) = 1$, the function is said to satisfy the strict avalanche criteria (SAC).

5.2.8 Correlation Immune Boolean Functions

Let $0 \leq k \leq n$. The function $g(x)$ on $GF(2)^n$ is k^{th} order correlation immune if the following equation

$$\sum_{x\in GF(2)^n} (-1)^{g(x)\oplus\alpha.x} = 0, \qquad 1 \leq \mathbf{wt}(\alpha) \leq k, \tag{5.17}$$

is satisfied, where $\mathbf{wt}(\alpha)$ is the Hamming weight for a vector $\alpha \in GF(2)^n$. In term of Walsh spectrum, a function g is said to be correlation immune of order t, denoted by $CI(t)$, if the output of the function is statistically independent of the combination of any of its inputs. For the Walsh spectrum, it holds that $W_g(\overline{\omega}) = 0,\ 0 \leq \mathbf{wt}(\overline{\omega}) \leq k$.

5.2.9 Algebraic Immunity

The Boolean function obtained by the product of the truth table of two Boolean functions h and g by $h.g$. The algebraic immunity (AI) of a Boolean functions h on $GF(2)^n$ is defined as the lowest degree of the functions g from $GF(2)^n$ to $GF(2)$ for which $h.g = \overline{0}$ or $(h+\overline{1}).g = \overline{0}$. The function g for which $h.g = \overline{0}$, is called an annihilator of h. It has been shown by Courtois in [121] that $AI(h) \leq \lceil \frac{n}{2} \rceil$. The algebraic immunity of an S-box, denoted by AI , is defined as

$$\Gamma = ((t-r)/n)^{[(t-r)/n]}, \tag{5.18}$$

where is the number of equations it satisfies and is the number of monomials in these equations [[122, 123]]. The algebraic immunity of an S-box depends on the number and type of linearly independent multivariate equations it satisfies.

5.2.10 Transparency Order of S-Boxes

Since Kocher firstly introduced the side-channel attack in 1996 [124], a lot of such attacks have been reported on a wide variety of cryptographic implementations. Among these attacks, the DPA is one of the most powerful attacks on iterated block ciphers. In [125], Prouff revised DPA attacks in terms of correlation coefficients between two Boolean functions for the hamming weight power consumption model and introduced a new characteristic called transparency order (TO) for S-Boxes in block ciphers. This metric is to quantify the resistance of S-Boxes against DPA attack. The TO of an $n \times m$ S-Box is defined as follows:

$$T_s = \max_{\beta \in GF(2)^n} \left(|m - 2\mathbf{wt}(\beta)| - \frac{1}{2^{2^n} - 2^n} \sum_{a \in GF(2)^n} \left| \sum_{v \in GF(2)^n, \mathbf{wt}(v)=1} (-1)^{v.\beta}_{D_a S} W(0,v) \right| \right). \tag{5.19}$$

Some properties of TO are studied in [125]. It has been proved that the smaller the TO of an S-Box, the higher its resistance would be against the DPA attacks. The trivial but tight upper bound and the lower bound on the TO of an (n,m) function are $0 \leq T_s \leq m$.

Table 5.2: Comparative analyses of proposed chaotic S-boxes with existing block ciphers.

Algebraic Properties	Proposed	AES	APA	Gray	Prime	S_8-AES	Skipjk	Xyi
Balancedness	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Algebraic degree	7	7	7	7	7	7	7	7
Algebraic immunity	3	4	4	4	4	4	4	4
Correlation immunity	0	0	0	0	0	0	0	0
Transparency order	7.766	7.860	7.859	7.860	7.756	7.857	7.821	7.822

We have analyzed the strength of our proposed chaotic S-box by placing comparative study among the existing S-boxes. The balancedness of the proposed chaotic block cipher suggest that information is normally distributed which is an important characteristics for a block cipher to be secure against different attacks. The algebraic degrees and correlation immunity for each S-box (see table 5.2) are same which shows that proposed chaotic S-box quality resistance against algebraic and correlation attacks. The algebraic immunity and transparency order of the projected and existing S-boxes are closely rated. These analyses are another way to study the strength of block ciphers for image and video encryption applications.

5.3 Security Analysis

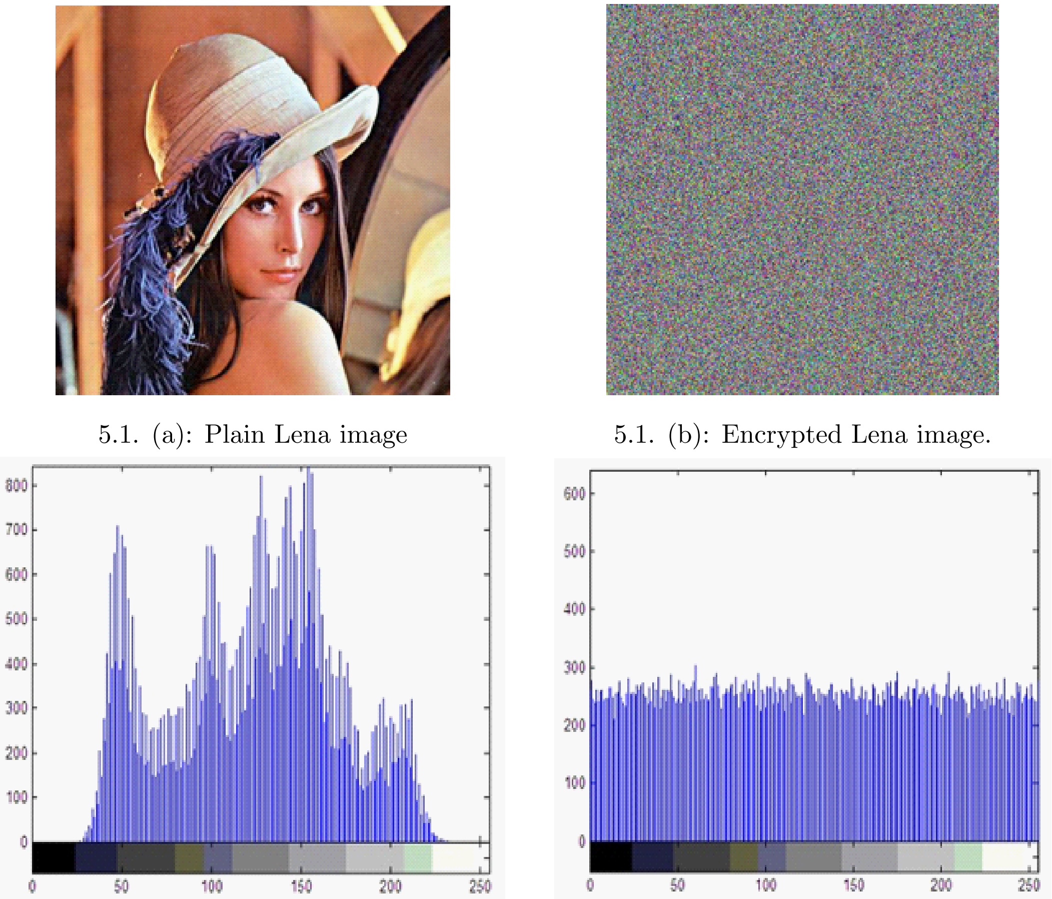
In order to test the performance of the proposed nonlinear component and its application to image encryption, we analyze the S-box transformation process with the application of statistical tests. The tests determine resistance to cryptanalysis by statistically analyzing the inputs and outputs. A detailed description of these tests is presented as follows.

5.3.1 Statistical Analysis

The statistical analyses are good source of determining resistance against brute force attacks, approximation attacks and differential attacks. The histogram analysis provides basic information about the distribution of pixel values after the nonlinear transformation. The effects of substitution and permutation can be observed with this analysis.

Histogram Analysis

In order to measure the similarities pertaining to statistical data, the histogram analysis provides insight into the changes caused by the substitution process. The distribution of pixel values or intensity levels after encryption is analyzed to estimate the amount of uniformity in all the region of the image. The original picture of sample image of Lena of size 256×256 is shown in Fig. 5.1(a). The encrypted version of this picture is seen in Fig. 5.1(b) and it is observed that the texture of the image is highly diffused. The analysis of the histogram also yields a flat response of the distribution as seen in Fig. 5.1(c) and Fig. 5.1(d). The distribution depicted in Fig. 5.1(d) is close to uniform distribution, which shows good diffusion after the nonlinear transformation. While the histogram is a basic test to measure the distribution of samples in the encrypted image, the entropy test provides deeper insight into the randomness of pixel values after the transformation.



5.1. (a): Plain Lena image

5.1. (b): Encrypted Lena image.

5.1. (c): Histogram of Plain Lena image.

5.1. (d): Histogram of encrypted Lena image.

Information Entropy

The degree of uncertainty is measured by the information entropy test. Higher entropy values in an image yields more randomness that makes the perception of artifacts more difficult [119]. The entropy in an image is determined as,

$$H(m) = -\sum_{i=0}^{2^m-1} p(m_i)\log_2(p(m_i)), \tag{5.20}$$

where $p(m_i)$ represents the probability of the occurrence of a pixel value m_i. The entropy of an image is represented in bits.

Table 5.3: The information entropy between plaintext and ciphertext.

Different images	Entropy	Ref. [109]
Plain image	7.2713	7.2713
Ciphered image	7.9972	7.9854

A good random image data has entropy of 8. In various images, the entropy never reaches the maximum value. After encryption, if the image encrypted data is highly random, the entropy must approach the maximum limit of 8. If the entropy of image samples drops below 8, there is a possibility to perceive image and the results of cryptanalysis may yield some information. In Fig. 5.1, the distribution is estimated statistically and entropy is measured. It is seen that the experimental values are very close to the ideal value of 8, which shown the security strength of the proposed nonlinear transformation algorithm.

Correlation Coefficient Analysis

The histogram analysis and entropy analysis show the global characteristics of randomness of an image. The adjacent pixels must be processed to determine locally the properties exhibited by the texture before and after encryption [126]. The correlation between two pixels in different directions is calculated by Eq. 5.21. A low correlation between adjacent pixels yields good encryption image with desirable resistance properties. The correlation coefficient is calculated as,

$$r = \frac{Cov(X,Y)}{S.D(X) \times S.D(Y)}, \tag{5.21}$$

where

$$\begin{aligned} Cov(X,Y) &= \sigma_{XY} = E[XY] - E[X]E[Y],\ S.D(X) = \sqrt{E[X^2] - (E[X])^2}, && (5.22) \\ S.D(Y) &= \sqrt{E[Y^2] - (E[Y])^2}, E[X] = \sum_{i=1}^{N} x_i p(x_i)\ , E[Y] = \sum_{j=1}^{N} y_j p(y_j). && (5.23) \end{aligned}$$

where $Cov(X,Y)$ is covariance of random variables X and Y, $E(X), E(Y)$ are expected value of and X, Y, $S.D(X)$, $S.D(Y)$ are standard deviation of random variables and respectively. The intensity values of adjacent pixels are represented by X and Y and the number of selected adjacent pixels are given as N. In the experiments, 1000 pairs of neighboring pixels are randomly chosen to calculate the correlation. The results of the correlation analysis are shown in Table 5.4.

Table 5.4: The related coefficient between plaintext and ciphertext.

Different images	Vertical direction	Horizontal direction	Diagonal direction
Plain image	0.9674	0.9119	0.8753
Ciphered image	0.0107	0.0141	0.0097
Ciphered image [110]	0.0383	0.0430	0.0117

5.3.2 Differential Analysis

The encryption process must be highly sensitive to small changes at the input. A minute change at the input must cause substantial shift in the output texture. For example, a slight modification in the value of one pixel must result in large number of changes in the encrypted image pixels. The effectiveness of differential cryptanalysis diminishes while the difficulty level increases in determining the relationship between input and output image. In this paper, we apply some prevailing differential analysis, which include: mean of absolute error (MAE), number of pixels change rate (NPCR) and unified average changing UACI [127, 128]. These tests are discussed as follows:

Mean Absolute Error

The grey levels in the images are quantized and represented as $C(i,j)$. In addition, $P(i,j)$ be represents the gray levels acquired by individual pixels at a particular location. The dimensions of the cipher are $W \times H$. The two images are compared and the resulting MAE is defined as,

$$MAE = \frac{1}{W \times H} \sum_{j=1}^{W} \sum_{i=1}^{H} \left| C(i,j) - P(i,j) \right| . \tag{5.24}$$

NPCR Analysis

In this analysis, we consider two encrypted images whose source images only differ by one pixel. If the first image is represented by $C_1(i,j)$ and the second as $C_2(i,j)$, the NPCR is evaluated as,

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\ \%, \tag{5.25}$$

where $D(i,j)$ is defined as

$$D(i,j) = \begin{cases} 0, & if \;\; C_1(i,j) = C_2(i,j), \\ 1, & if \;\; C_1(i,j) \neq C_2(i,j). \end{cases} \tag{5.26}$$

UACI Analysis

The UACI analysis is mathematically represented as,

$$UACI = \frac{1}{W \times H} \sum_{j=1}^{W} \sum_{i=1}^{H} \left[\frac{|C(i,j) - P(i,j)|}{255} \right] \times 100\ \% \tag{5.27}$$

In this work, we perform tests on a sample image of dimension 256×256 with 256 levels of gray. The results of MAE are shown in Table 5.5 where the performance is seen with fluctuation between rows and columns. The encryption performance increase with larger values of MAE results. The outcome of other two tests, NPCR and UACI are shown in Table 5.5. The NPCR analysis shows response to changes of

0.01% in the input images. In addition, the UACI show the response to a change in one pixel, which is very low. A rapid change in the original image show little changes in the resulting encrypted image. The results of these three tests are shown in Table 5.5.

Table 5.5: Sensitivity to plaintext and MAE.

Images	NPCR			Ref. [129]	UACI			Ref. [129]	MAE
	Max.	Min.	Average		Max.	Min.	Average		
Lena	99.68	99.54	99.6124	99.5860	33.71	33.28	33.4591	33.4190	77.35
Baboon	99.66	99.56	99.6124	-	33.66	33.32	33.4891		73.91
Peppers	99.66	99.55	99.6124	-	33.68	33.27	33.5057		74.71

5.4 Conclusion

In this work, we have proposed an algorithm to construct nonlinear components used in image encryption applications. The chaotic Boolean bit function is employed in the construction of nonlinear substitution components that are applied to the encryption of images. The random nature of chaotic system is fully exploited in embedding their properties in the cryptographic system. The simulation results show high resistance to brute force attacks, linear and differential cryptanalysis. The proposed system is suitable for high speed communication channels that carry extensive image and video data.

Chapter 6

An Efficient Chaotic Image Encryption Scheme

In this communication, we have presented a technique to synthesize resilient nonlinear mechanisms for the construction of substitution box for image encryption that utilize a multiplicative group of nonzero elements of Galois field of order 256. The proposed nonlinear component assists in transforming the intelligible message or plaintext into an enciphered format by the use of exponential and Tinkerbell chaotic maps. The proposed substitution box is sensitive to the initial conditions provided to the chaotic system, which are subsequently used as parameters in creating an instance. The simulation results show that the use of the proposed substitution box to image encryption scheme provides an efficient and secure way for real-time communications.

6.1 Exponential Chaotic Map

When we design S-box, it is very important to find a proper permutation which has good properties in cryptology. We choose the following function . Let $g : N \to N$ defined as:

$$x \mapsto \begin{cases} g^x \ \text{mod} \ 257, & if \ x < 256 \\ 0, & if \ \ x = 256 \end{cases} \tag{6.1}$$

where $x = g^x \pmod{257}$ and $x \in N = \{0, 1, 2, ..., 255\}$. We select g as a primitive element which generates the multiplicative group of nonzero elements of Galois field of order 256. There are 128 different values of g. In this case the mapping $x \mapsto g^x \pmod{257}$ is bijective. The $\mathbb{Z}^*_{257}$ is a multiplicative group of order $\varphi(257) = 256$, where 257 is a prime number, φ is the Euler Totient function and $\varphi(m)$ is equal to the number of integers in the interval $[1, m]$ which are relative prime to m. The order of an element $a \in \mathbb{Z}^*_p$ is the least positive integer t such that $a^t = 1(\text{mod} \ \ p)$. By Fermat's Little Theorem, we know that, if $gcd(a,p) = 1$, and p is a prime number, then $a^{p-1} = 1 (\text{mod } p)$. Thus by Lagrange's theorem, we also know that the order of 45 divides the group order i.e., 256 and thus the order of 45 must be a power of 2. We observe that so that the smallest integer (being a power of 2) such that is 256. Therefore, the order of 45 is equal to the group order, which proves that g is the generator of the group. Thus, the function $x \mapsto 45^x (\text{mod } 257)$, is a bijection from $\{0,1,2,...,255\}$ to $\{1,2,3,...,256\}$.

6.2 Algebraic Expression of the Proposed S-box

In this section, we are mainly discussed the algebra of proposed S-box. The following are main steps in constructing proposed S-boxes [136]-[137]:

- Take the multiplicative inverse in the finite field $\mathbb{Z}^*_{257}$; the element 256 is mapped to 0.
- The multiplicative inversion operation in the construction of S-box is the inversion $\mathbb{Z}^*_{257}$ in with the extension $256 \mapsto 0$. We define the following function $F(x)$ in $\mathbb{Z}^*_{257}$ corresponding to this multiplicative inversion step:

$$F(x) = \begin{cases} x^{-1}, & if \ x < 256 \\ 0, & if \ x = 256. \end{cases} \tag{6.2}$$

Since we can rewrite $x^{-1} = x^{2^s-1} = x^{255}$ for $x \neq 0 \in \mathbb{Z}^*_{257}$, we can rewrite as follows:

$$F(x) = x^{255}. \tag{6.3}$$

We decompose the affine transformation step in proposed S- box construction into two consecutive functions. Let $L_A(x)$ be a linear transformation in $GF(2^8)$ which can be expressed as follows:

$$y = L_A(x), \tag{6.4}$$

where

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}, \tag{6.5}$$

with x_i is the ith bit of the byte (is the LSB) and y_i is the ith bit of the byte y. As the permutation

$L_A(x)$ is a linear map $\mathbb{Z}_2$, it can be expressed as a linearized polynomial [23] with 8 terms:

$$L(x) = \sum_{i=0}^{7} \lambda_i x^{2^i}. \tag{6.6}$$

The final sub-step in AES S-box construction is the addition with the constant values $\{63\}$. We define the affine transformation function $H(x)$ in $GF(2^8)$:

$$H(x) = x \oplus d.$$

The proposed S-box is the combination of the power function $F(x)$, the linear transformation , $L_A(x)$ and the affine transformation $H(x)$:

$$S - box = H \circ L_A \circ F = H(L_A(F)) = L_A(x^{-1}) \oplus d, \tag{6.7}$$

where

$$L_A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \ d = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \tag{6.8}$$

The linearized polynomial of any linear permutation $L_A(x)$ over $GF(2^8)$ has at most eight terms. Therefore, if we substitute $L_A(x)$ by another linear permutation over $GF(2^8)$ and/or change the constant $\{63\}$

in $H(x)$ by another value in $GF(2^8)$. The proposed S-box is presented in Table 6.1.

Table 6.1: The proposed chaotic S-box.

2	242	113	1	64	88	185	190	17	220	236	47	240	49	9	14
55	104	43	155	102	50	83	10	135	223	56	181	72	28	227	186
189	152	60	168	172	39	109	107	13	5	18	134	197	151	221	120
199	215	213	149	157	211	243	184	119	103	251	187	45	99	67	171
91	68	167	148	84	165	212	92	117	244	23	36	228	182	100	114
226	115	65	235	207	153	245	222	139	195	111	248	225	41	219	110
75	70	231	69	23	133	159	147	214	104	87	44	170	241	140	35
128	112	200	54	127	93	188	130	48	192	230	37	22	237	146	145
137	247	158	90	141	79	179	176	57	71	46	234	61	97	3	0
62	232	125	105	77	32	194	166	142	198	205	217	253	144	209	136
76	233	180	129	106	196	94	53	95	89	4	175	218	116	238	27
101	30	163	178	121	150	96	202	118	174	19	156	201	255	208	122
126	224	51	73	6	239	210	58	206	80	131	249	40	193	252	138
143	15	98	254	25	12	66	250	161	33	11	78	169	31	81	74
7	38	164	29	42	82	16	21	183	8	20	173	154	124	160	59
162	123	24	177	132	86	229	203	63	85	191	216	52	34	246	108

6.3 Chaotic Sequence for Image Encryption

For generating the initial condition method described in [130] is used. Calculate two parameters c_1 and c_2 as in Eqs. (6.9) – (6.10) :

$$\begin{aligned} c_1 &= \frac{1}{2^8} \bmod \left(\sum_{i=1}^{m/2} \sum_{j=1}^{n} P_{ij}, 2^8 \right), && (6.9) \\ c_2 &= \frac{1}{2^8} \bmod \left(\sum_{i=m/2}^{m} \sum_{j=1}^{n} P_{ij}, 2^8 \right), && (6.10) \end{aligned}$$

where P_{ij} is the value of the image pixel at location (i, j) in the image. Additional let $x'_0 = 0.59$ and $y'_0 = 0.15$. Compute initial conditions as in Eqs. (6.11) – (6.12)

$$\begin{aligned} x_0 &= \bmod\left[(x'_0 + c_1), 1\right], && (6.11) \\ y_0 &= \bmod\left[(y'_0 + c_2), 1\right]. && (6.12) \end{aligned}$$

The proposed algorithm uses Tinkerbell map based on chaotic sequence is defined as in Eqs. (6.13) – (6.14) :

$$\begin{aligned} x_{n+1} &= x_n^2 - y_n^2 + ax_n + by_n, & (6.13) \\ y_{n+1} &= 2x_ny_n + cx_n + dy_n, & (6.14) \end{aligned}$$

where a, b, c and d are non-zero parameters which are the part of secret key. For parameter values $a = 0.90, b = -0.6013, c = 2.0$ and $d = 0.50$, we get the chaotic attractor of this map. Such a chaotic motion gets controlled and display regular behavior for $a = 0.90, b = -0.6, c = 2.0$ and $d = 0.27$ and keeping other parameters same. Use x_0 and y_0 as the initial approximation for Eqs. (6.11) – (6.12) and obtain two matrices of size 1×256 as in Eqs. (6.13) – (6.14):

$$\{X_i = (x_1, x_2, x_3, ..., x_i),\ Y_i = (y_1, y_2, y_3, ..., y_i)\}\,. \tag{6.15}$$

Now for permuting the rows and columns, we will use the following relation given below:

$$\begin{aligned} R(i) &= R((X_i \times m) \mod i), & (6.16) \\ C(j) &= C((Y_i \times m) \mod j). & (6.17) \end{aligned}$$

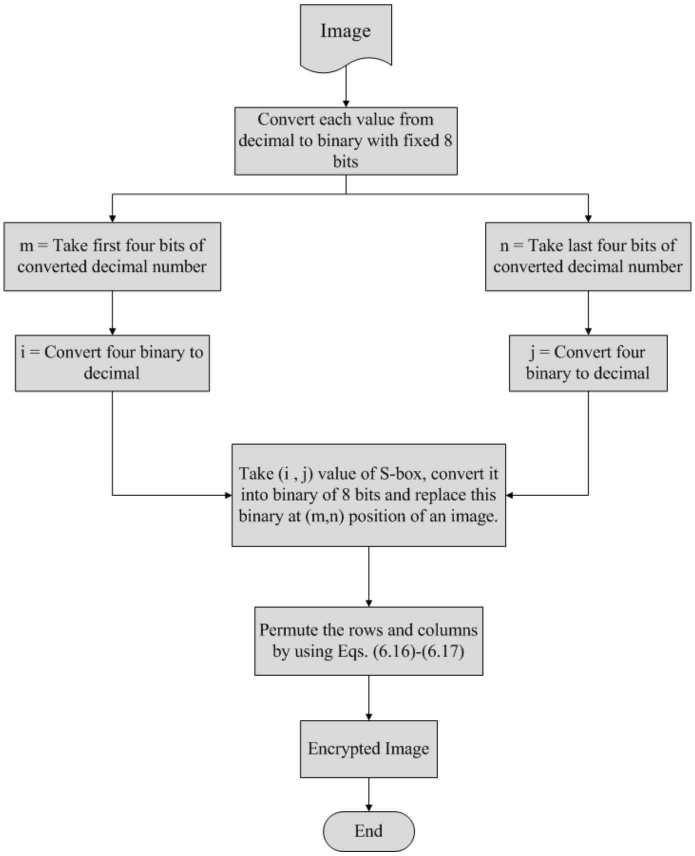


Image
Convert each value from decimal to binary with fixed 8 bits
m = Take first four bits of converted decimal number
n = Take last four bits of converted decimal number
i = Convert four binary to decimal
j = Convert four binary to decimal
Take (i , j) value of S-box, convert it into binary of 8 bits and replace this binary at (m,n) position of an image.
Permute the rows and columns by using Eqs. (6.16)-(6.17)
Encrypted Image
End

Fig. 6.1: Flow diagram for proposed chaotic image encryption.

6.4 Statistical analyses

The statistical analyses provide insight into the working of any cryptographic system. In order to evaluate the performance of the proposed S-box, we conduct histogram analysis, correlation analysis, mean square error, peak signal to noise ratio, encryption quality, entropy and sensitivity analyses which includes, mean absolute error (MAE), number of pixel changing rate (NPCR) and unified average changed intensity (UACI). The results of correlation analysis show the extent of similarity between the original and encrypted data. If there are any traces of correlation, there is a possibility that cryptanalysis my decipher the original data or may be able to partially interpret information. The mean squared error (MSE) allow us to compare the pixel values of original image to encrypted image. The MSE represents the average of the squares of the errors between actual image and ciphered image. The error is the amount by which the values of the original image differ from the encrypted image. The PSNR computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and an encrypted image. The higher value of PSNR indicates better quality of the image encryption. With the application of encryption to a picture change happens in pixels values as contrasted with those qualities before encryption. Such change may be unpredictable. This implies that the higher the change in pixels values, the more successful will be the picture encryption and subsequently the encryption quality. So the encryption quality may be communicated as far as the aggregate changes in pixels values between the first picture and the scrambled one. A measure for encryption quality may be communicated as the deviation between the plain image and encoded image.

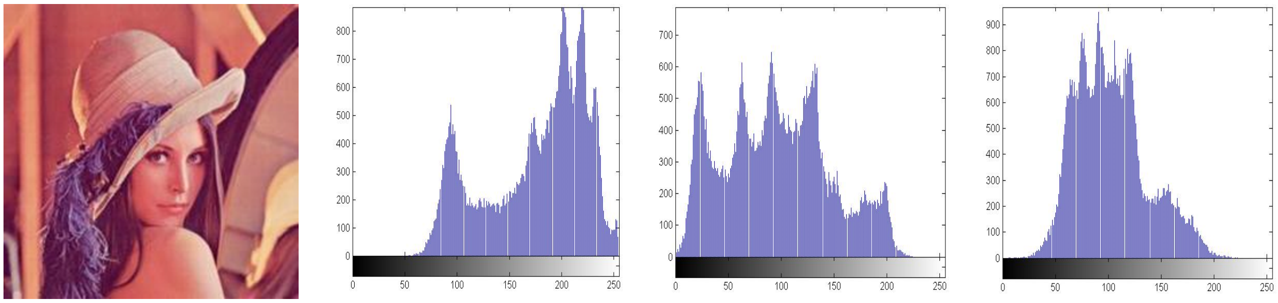
In the entropy analysis, we determine the amount of randomness introduced in the plaintext. This measure is also useful in image encryption application were visual form of data may provide additional information about the original data. As a rule, an alluring trademark for a scrambled image is continuously touchy to the little changes in plain-image (e.g. changing only one pixel). Enemy can make a little change in the information picture to watch changes in the result. By this system, the serious relationship between original image and cipher image can be found. In the event that one little change in the plain-image can result in a significant change in the cipher-image, with respect to diffusion and confusion, then the differential attack really loses its productivity and gets to be useless. There are three basic measures were utilized for differential analysis: MAE, NPCR and UACI. The greater the MAE value, the better the encryption security. NPCR implies the number of pixels change rate of encoded picture while one pixel of plain-image is changed. UACI which is the unified average changing intensity, measures the normal power of the contrasts between the plain-image and encrypted image. We discuss in detail the implementation and analysis of the tests used to benchmark the performance of the proposed S-box.

6.4.1 Histogram

One of the best outstanding features for measuring the security of image encryption systems is uniformity of the image's histogram of encrypted images [138]. We took six color images with size of 256×256 that have different contents and their histograms are calculated. The histogram of plain-images comprises huge sharp rises followed by sharp declines and the histogram of all cipher-images under the suggested procedure is equally identical and meaningfully diverse from that of the plain-images, which makes statistical assaults tough (see Figs. $6.2 - 6.13$). Hence it does not provide any clue to be employed in a statistical analysis attack on the encrypted image. The equation used to calculate the uniformity of a histogram caused by the proposed encryption scheme is justified by the chi-square test as follows:

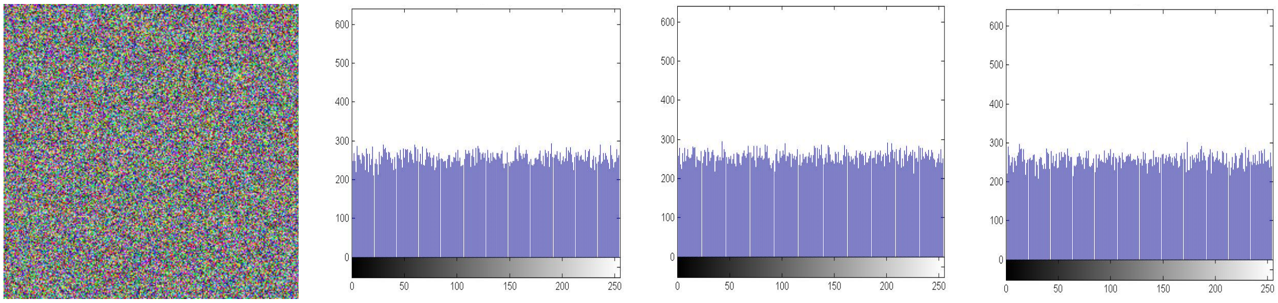
$$\chi^2 = \sum_{j=1}^{256} \frac{(f_0 - f_e)^2}{f_e}, \tag{6.18}$$

where j is the number of gray levels (256), f_0 is the observed occurrence frequencies of each gray level $(0 - 255)$, and f_e is the expected occurrence frequency of each gray level while $f_e = M \times N/256$, M and N are the height and width of the plain/cipher image, respectively. Hence, f_e is equal to 256 for an image size of 256×256. The lower value of the chi-square test indicates a better uniformity. Assuming a significant level of 0.05, $\chi^2_{(255,0.05)} = 293.2478$. Chi-square value for the final encrypted Lena image of the proposed system is 195.32 i-e., $\chi^2_{(test)} = 195.32$. This implies that the null hypothesis is not rejected and the distribution of the encrypted histogram is uniform $\chi^2_{(test)} < \chi^2_{(255,0.05)}$. The chi-square values of plain-images and cipher-images are shown in Table 6.2.



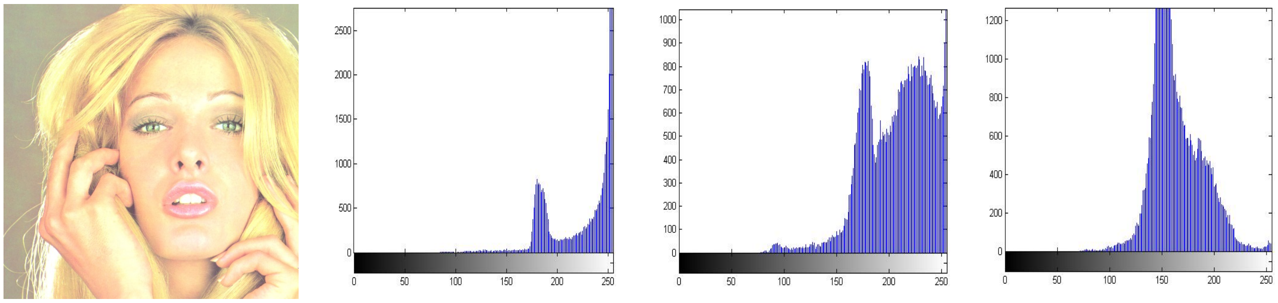
(a) (b) (c) (d)

Fig. 6.2: (a) Lena Image; (b) Histogram of Lena image for red component of Lena image (c) Histogram of Lena image for green component of Lena image (d) Histogram of Lena image for blue component of Lena image.



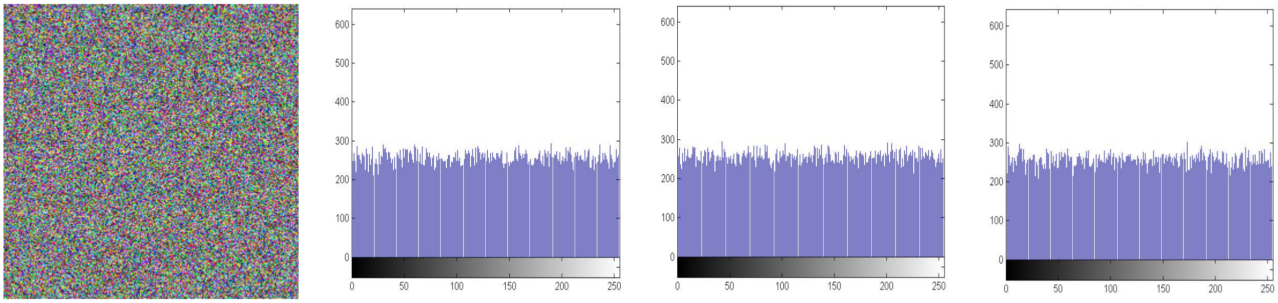
(a) (b) (c) (d)

Fig. 6.3: (a) Lena encrypted image; (b) Histogram of Lena encrypted image for red component of Lena image (c) Histogram of Lena encrypted image for green component of Lena image (d) Histogram of Lena encrypted image for blue component of Lena image.



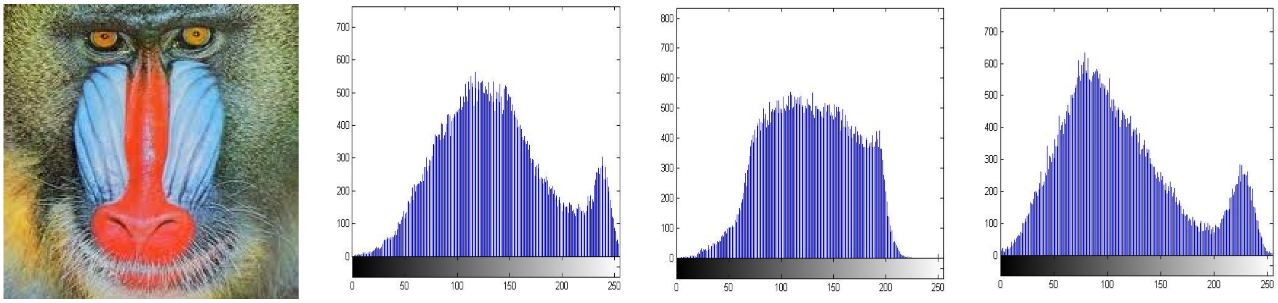
(a) (b) (c) (d)

Fig. 6.4: (a) Tiffany Image; (b) Histogram of Tiffany image for red component of Tiffany image (c) Histogram of Tiffany image for green component of Tiffany image (d) Histogram of Tiffany image for blue component of Tiffany image.



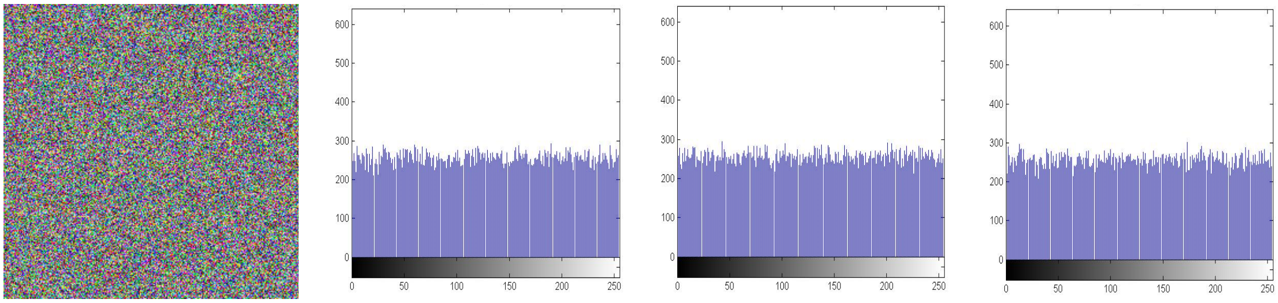
(a) (b) (c) (d)

Fig. 6.5: (a) Tiffany encrypted Image; (b) Histogram of Tiffany encrypted image for red component of Tiffany image (c) Histogram of Tiffany encrypted image for green component of Tiffany image (d) Histogram of Tiffany encrypted image for blue component of Tiffany image.



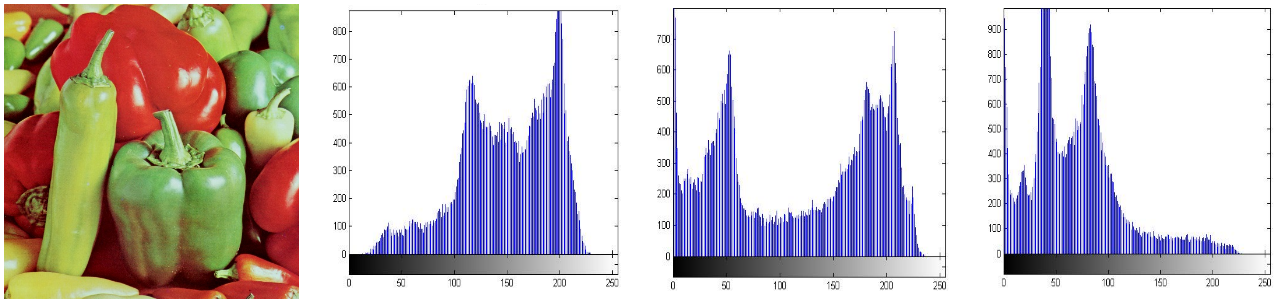
(a) (b) (c) (d)

Fig. 6.6: (a) Baboon Image; (b) Histogram of Baboon image for red component of Baboon image (c) Histogram of Baboon image for green component of Baboon image (d) Histogram of Baboon image for blue component of Baboon image.



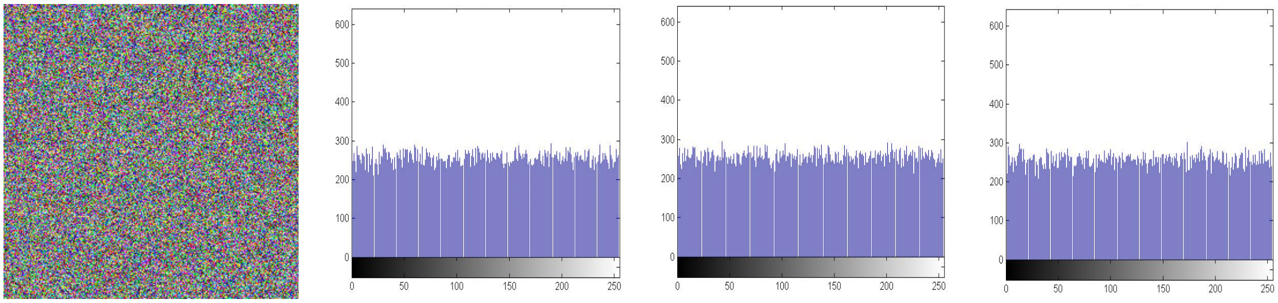
(a) (b) (c) (d)

Fig. 6.7: (a) Baboon encrypted Image; (b) Histogram of Baboon encrypted image for red component of Baboon image (c) Histogram of Baboon encrypted image for green component of Baboon image (d) Histogram of Baboon encrypted image for blue component of Baboon image.



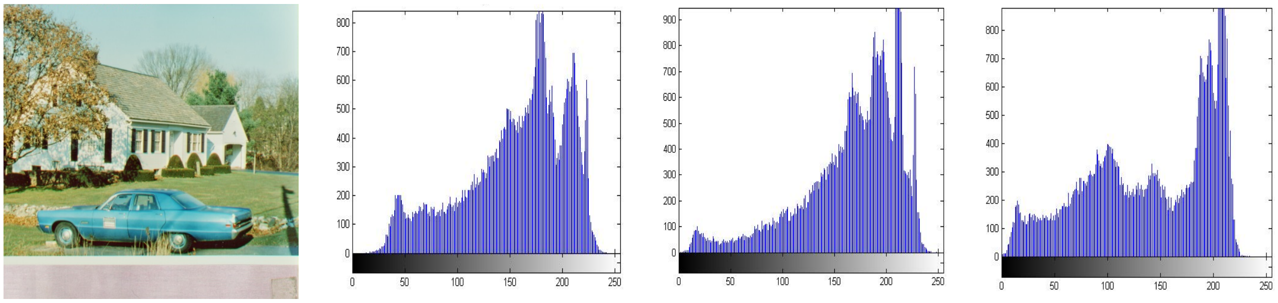
(a) (b) (c) (d)

Fig. 6.8: (a) Pepper Image; (b) Histogram of Pepper image for red component of Pepper image (c) Histogram of Pepper image for green component of Pepper image (d) Histogram of Pepper image for blue component of Pepper image.



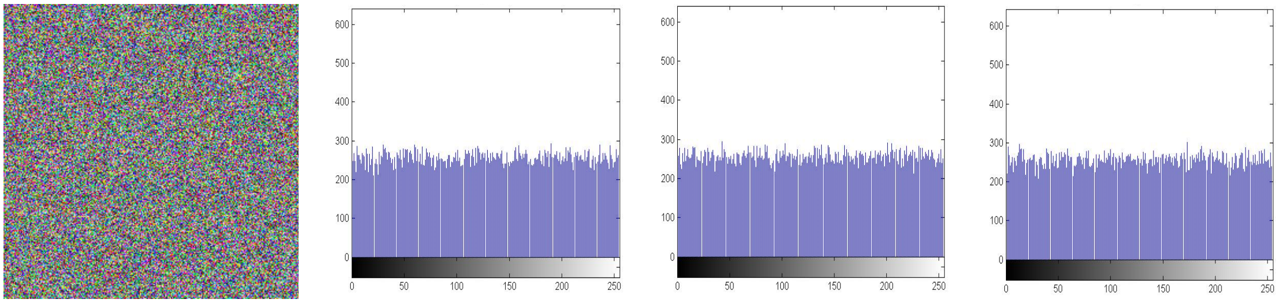
(a) (b) (c) (d)

Fig. 6.9: (a) Pepper encrypted Image; (b) Histogram of Pepper encrypted image for red component of Pepper image (c) Histogram of Pepper encrypted image for green component of Pepper image (d) Histogram of Pepper encrypted image for blue component of Pepper image.



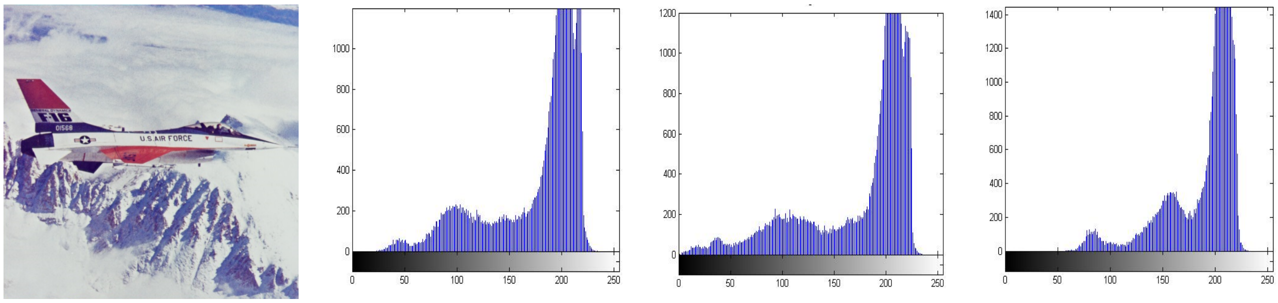
(*a*) (*b*) (*c*) (*d*)

Fig. 6.10: (a) House Image; (b) Histogram of House image for red component of House image (c) Histogram of House image for green component of House image (d) Histogram of House image for blue component of House image.



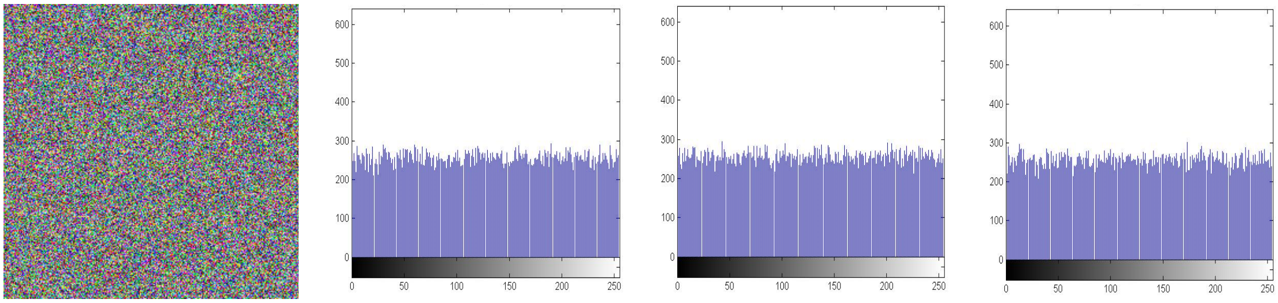
(*a*) (*b*) (*c*) (*d*)

Fig. 6.11: (a) House encrypted Image; (b) Histogram of House encrypted image for red component of House image (c) Histogram of House encrypted image for green component of House image (d) Histogram of House encrypted image for blue component of House image.



(*a*) (*b*) (*c*) (*d*)

Fig. 6.12: (a) Airplane Image; (b) Histogram of Airplane image for red component of Airplane image (c) Histogram of Airplane image for green component of Airplane image (d) Histogram of Airplane image for blue component of Airplane image.



(a) (b) (c) (d)

Fig. 6.13: (a) Airplane encrypted Image; (b) Histogram of Airplane encrypted image for red component of Airplane image (c) Histogram of Airplane encrypted image for green component of Airplane image (d) Histogram of Airplane encrypted image for blue component of Airplane image.

6.4.2 Correlation

It is important to determine the similarity between the original image and the encrypted image. This measure is useful for image encryption applications where the cryptanalysis has an additional advantage of visually perceiving the encrypted image and extracting unauthorized information. This analysis is performed in three different steps, in which the correlation between adjacent pixels in horizontal diagonal and vertical directions is evaluated. The selected pairs of pixels in horizontal, diagonal and vertical directions are processed for correlation in random locations in the data. Finally, the all the pixels are processed together to see the global perspective. These three cases are presented as:

Case 1: In this step, we select adjacent pixels (typically two) in horizontal and vertical directions from original and encrypted image and evaluate the coefficients. The Table 6.2, shows the results from this test that show considerable reduction in correlations between the two images.

Case 2: The pixels located diagonally in an image are processed to see the correlation between closely located pixels. A random selection of approximately 1000 pair of pixels, located in diagonal directions, is processed to determine the correlation.

Case 3: All the pixels are represented by two variables X and Y, which is the global representation of the entire image. The correlation for this entire set of pixels is calculated as [119]:

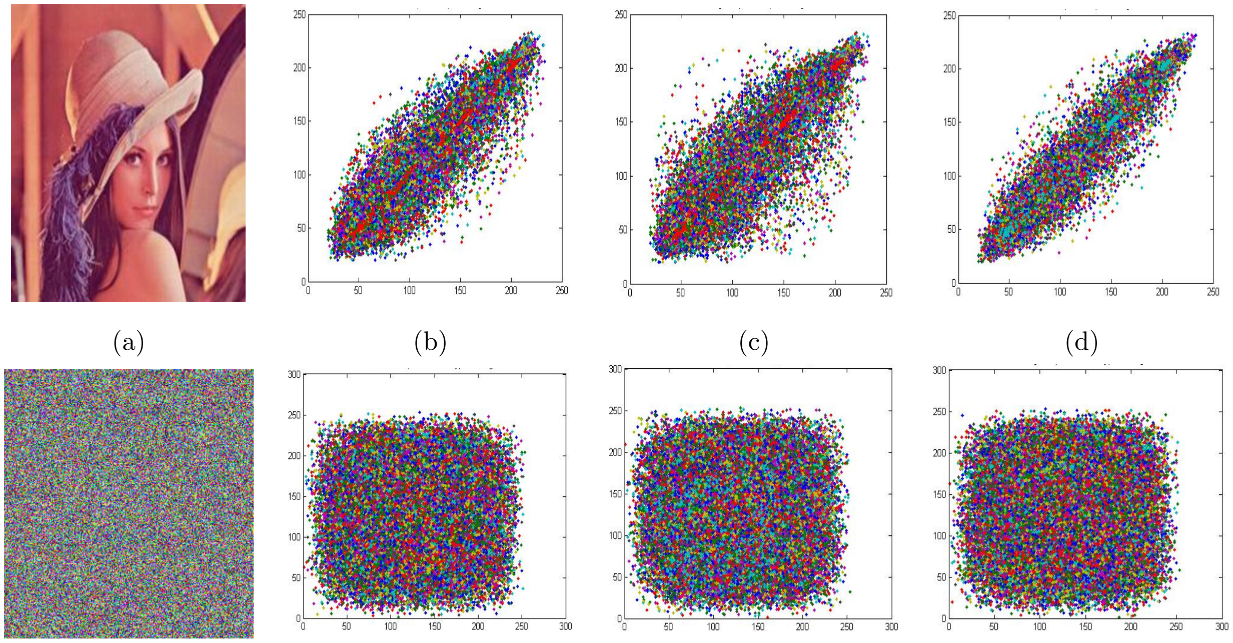
$$r_{x,y} = \frac{\sigma_{x,y}}{\sqrt{\sigma_x^2 \sigma_y^2}}, \tag{6.19}$$

where $\sigma_{x,y}$ is covariance of random variables X and Y, μ_X, μ_Y are expected value of X and Y; and σ_x^2, σ_y^2 are variances of random variables X and Y respectively. Each term is defined as follows:

$$\sigma_{x,y} = \sum_{j=1}^{N} (X_j - \mu_X)(Y_j - \mu_Y)/N, \quad \sigma_x^2 = \sum_{j=1}^{N} (X_j - \mu_X)^2/N, \tag{6.20}$$

$$\sigma_y^2 = \sum_{j=1}^{N}(Y_j - \mu_Y)^2/N, \quad \mu_x = \sum_{j=1}^{N} X_j/N, \quad \mu_y = \sum_{j=1}^{N} Y_j/N. \tag{6.21}$$

Finally Fig. 6.14, shows the correlation distribution of two horizontally adjacent pixels in the plain-image and that in the ciphered image. It is quite evident from the analyses of these correlation images that the proposed algorithm is capable of breaking the correlation among the pixels in neighboring which is astonishing achievement of anticipated scheme.



(a) (b) (c) (d)

(e) (f) (g) (e)

Fig. 6.14: Correlation of two adjacent pixels: (a) Plain-Lena image, (b) Distribution of two horizontally adjacent pixels in the plain-Lena image, (c) Distribution of two vertically adjacent pixels in the plain-Lena image, (d) Distribution of two diagonally adjacent pixels in the plain-Lena image, (e) Encrypted-Lena image, (f) Distribution of two horizontally adjacent pixels in the encrypted-Lena image, (g) Distribution of two vertically adjacent pixels in the encrypted-Lena image, and (h) Distribution of two diagonally adjacent pixels in the encrypted- Lena image.

Table 6.2: Chi-square test and correlation coefficient and of different plain-image and cipher-image.

	Plain image				Encrypted image			
Images	Chi-square	Correlation coefficient			Chi-square	Correlation coefficient		
	values	Horizontal	Diagonal	Vertical	values	Horizontal	Diagonal	Vertical
Lena	28588	0.9268	0.9068	0.9604	195.32	0.00091	0.0021	-0.0007
Tiffany	133363	0.8889	0.8476	0.9266	257.23	-0.0079	0.0008	0.0005
Baboon	44395	0.6935	0.6086	0.5963	235.79	-0.0001	0.0003	0.0009
Peppers	36778	0.9455	0.8951	0.9407	240.57	0.00076	-0.0012	0.0001
House	42952	0.9328	0.8898	0.9288	249.67	0.00091	0.0051	0.0001
Airplane	163822	0.9048	0.8309	0.8940	241.52	0.00071	0.0003	-0.0005

6.4.3 Mean Square Error

To evaluate the reliability of the proposed algorithm, mean square error (MSE) between encrypted image and original image is measured. MSE is calculated using the following equation [118]:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i,j) - C(i,j))^2, \tag{6.22}$$

where $M \times N$ is the size of the image. The parameters $P(i,j)$ and $C(i,j)$ refer to the pixels located at the ith row and the jth column of original image and encrypted image, respectively. The larger the MSE value, the better the encryption security (see Table 6.3).

6.4.4 Peak Signal to Noise Ratio

The encrypted image quality is evaluated using peak signal to noise ratio (PSNR) [118] which is described by the following expressions:

$$PSNR = 10 \log_2 \left(\frac{I_{\max}^2}{\sqrt{MSE}} \right), \tag{6.23}$$

where $I_{\max}$ is the maximum of pixel value of the image. The PSNR should be a low value which corresponds to a great difference between the original image and the encrypted image. The effectiveness of the proposed method, evaluated in terms of MSE and PSNR are tabulated in Table 6.3.

6.4.5 Encryption Quality

Plain-image pixels' gray levels change after image encryption as compared to their original values before encryption. This means that the higher the change in pixels' values, the more effective will be the image encryption and hence the encryption quality (EQ). The quality of image encryption may be determined as follows: let $C(i,j)$ and $P(i,j)$ be the gray value of the pixels in cipher and plain-image, each of size $M \times N$ pixels with L gray levels and $C(i,j),\ P(i,j) \in \{0, 1, 2, ..., L-1\}$. We will define

$H_L(P)$ and $H_L(C)$ as the number of occurrences for each gray level L in the plain-image and cipher-image, respectively. The EQ represents the average number of changes to each gray level L. The larger the EQ value, the better the encryption security (see Table 6.3). The EQ is calculated as:

$$EQ = \sum_{L=0}^{2^8-1} \left(H_L(C) - H_L(P)\right)^2 / 2^8. \tag{6.24}$$

6.4.6 Entropy

The texture of an image can be characterized by the measurement of entropy. This quantity is defined as:

$$H = -\sum_{j=0}^{N-1} p(x_j) \log_b p(x_j), \tag{6.25}$$

where a random variable X take n outcomes i-e.,$\{x_0, x_1, x_2, ..., x_n\}$, $p(x_j)$ is the probability mass function of x_j outcome and b is the base of the logarithm used. A benchmark for the entropy analysis is presented in Table 6.3. The results show that the performance of the proposed S-box better than some of the prevailing S-boxes used in image encryption applications [118].

Table 6.3: Statistical encryption quality parameters of proposed algorithm and its comparison.

Images	Projected technique				Ref. [131]	Ref. [132]	
	MSE	PSNR	Entropy	EQ	MSE	PSNR	Entropy
Lena	10351	9.5513	7.9979	150.12	7510	9.2322	7.9977
Tiffany	14160	8.5132	7.9977	293.43	-	-	-
Baboon	8053	9.3214	7.9974	195.96	6583	9.5466	7.9970
Peppers	9050	8.9455	7.9974	175.11	8298	8.9914	7.9973
House	10259	8.9931	7.9973	149.13	-	-	-
Airplane	11105	8.9192	7.9972	251.95	-	-	-

6.4.7 Sensitivity Analyses

Attackers often make a small change to the plain-image and use the proposed algorithm to encrypt the plain-image before and after this change. By comparing these two encrypted images they find out the relationship between the plain-image and the cipher-image. This kind of attack is called differential attack. In order to resist differential attack, a minor alternation in the plain-image should cause a substantial change in the cipher-image [133, 134]. To test the influence of one-pixel change on the whole image encrypted by the proposed algorithm, three common measures can be used: mean absolute error (MAE), number of pixels' change rate (NPCR) and unified average changing intensity (UACI).

Mean Absolute Error

The mean absolute error (MAE) is a criterion to examine the performance of resisting differential attack. Let $C(i, j)$ and $P(i, j)$ be the gray level of the pixels at the ith row and the jth column of an $M \times N$ cipher and plain-image, respectively. The MAE between these two images is defined as [118]:

$$MAE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| C(i,j) - P(i,j) \right| . \tag{6.26}$$

The larger the MAE value, the better the encryption security. The mean absolute error (MAE) is figured to measure how the cipher image $C(i, j)$ is not the same as the plain image $P(i, j)$.

Number of Pixel Changing Rate (NPCR)

In this analysis, we consider two encrypted images whose source images only differ by one pixel. If the first image is represented by $C_1(i, j)$ and the second as $C_2(i, j)$, the NPCR is evaluated as [135]:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \tag{6.27}$$

where $D(i, j)$ is defined as:

$$D(i,j) = \begin{cases} 0, & if\ C_1(i,j) = C_2(i,j), \\ 1, & if\ C_1(i,j) \neq C_2(i,j). \end{cases} \tag{6.28}$$

Unified Average Changed Intensity (UACI)

The UACI analysis is mathematically represented as,

$$UACI = \frac{1}{W \times H} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| \frac{C_1(i,j) - C_2(i,j)}{255} \right| \times 100\%. \tag{6.29}$$

In this work, we perform tests on a sample image of dimension 256 × 256 with 256 levels of gray. The results of MAE are shown in Table 6.4 where the performance is seen with fluctuation between rows and columns. The encryption performance increase with larger values of MAE results. The outcome of other two tests, NPCR and UACI are also shown in Table 6.4. The NPCR analysis shows response to changes of 0.01% in the input images. In addition, the UACI show the response to a change in one pixel, which is very low. A rapid change in the original image show little changes in the resulting encrypted image.

The results of these three tests are shown in Tables 6.4 and 6.5 respectively.

Table 6.4: Comparison of NPCR and UACI criteria of proposed method and the others.

Images	Projected technique		Ref. [135]
	NPCR	UACI	NPCR
Lena	99.6692	33.5051	99.60244
Baboon	99.6562	33.5571	-
Peppers	99.6626	33.4733	99.60352
Airplane	99.6492	33.4895	-

Table 6.5: Sensitivity to plaintext and MAE.

Images	NPCR			UACI			MAE
	Max	Min	Average	Max	Min	Average	
Lena	99.68	99.54	99.6124	33.71	33.28	33.4591	77.35
Tiffany	99.67	99.57	99.6124	33.72	33.29	33.5173	76.23
Baboon	99.66	99.56	99.6124	33.66	33.32	33.4891	73.91
Peppers	99.66	99.55	99.6124	33.68	33.27	33.5057	74.71
House	99.65	99.53	99.6124	33.67	33.31	33.5251	75.65
Airplane	99.66	99.54	99.6124	33.66	33.30	33.4931	74.31

6.5 Conclusion

In this chapter, an updated version of image encryption algorithm has been offered which is based on multiplicative group of nonzero elements of Galois field $\mathbb{Z}_{257}^{*}$, exponential and Tinkerbell chaotic maps. The experimental analysis and results demonstrate that the anticipated algorithm has desirable properties such as high sensitivity to a small change in plain-image, low correlation coefficients, low chi-square scores, high mean square values, low peak signal to noise ratio, high encryption quality and large information entropy. All these features verify that the proposed algorithm is robust and effective for image encryption. The NPCR and UACI scores show that proposed version is extremely subtle to a slight modification in the plain-image. Several other simulation analyses and comparative studies authenticate the enriched security performance of the suggested version.

Chapter 7

A Novel Image Encryption Technique Based on Hénon Chaotic Map and S_8 Symmetric Group

The structure of cryptographically resilient substitution boxes (S-boxes) plays a central role in devising safe cryptosystems. The design of chaos-based S-boxes by means of chaotic maps obtained more devotion in current ages. We have suggested novel S-boxes based on the chaotic maps and S_8 symmetric group. We have experimented our chaos-based S-box for image encryption applications and analyze its strength with statistical analyses.

7.1 Fundamental Properties of Chaotic Systems

Chaos has been witnessed in many natural structures covering a significant amount of technical and industrial areas. These occurrences display definite possessions that mark them difficult and volatile. Chaos theory deals with constructions that progress in time to a specific kind of dynamical actions. Several authors have addressed the mathematical theory of chaos due to its vast and most applicable effects in various fields of science. In broad spectrum, these schemes follow a definite set of procedures of improvement. Generally, chaos happens simply in certain deterministic nonlinear systems. Clearly, chaos seems when there is a continuous and disorganized looking long-term progression that fulfills definite mathematical benchmarks. There are certain set of properties that sum up the features witnessed in chaotic systems. These measured the mathematical principles that describe chaos (see Table 3.1).

7.1.1 Chaotic Hénon Maps

The Hénon map is proposed by the French astronomer and mathematician Michel Hénon [140]. The Hénon map has yielded a great deal of interesting characteristics as it was studied. At their core, the Hénon map is basically a family of functions defined from $f_{\alpha\beta} : \mathbb{R}^2 \to \mathbb{R}^2$ and denoted by:

$$f_{\alpha\beta}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y + 1 - \alpha x^2 \\ \beta x \end{pmatrix}, \tag{7.1}$$

where α and β are (positive) bifurcation parameters (see Figs. 7.1 – 7.2 and 7.3).

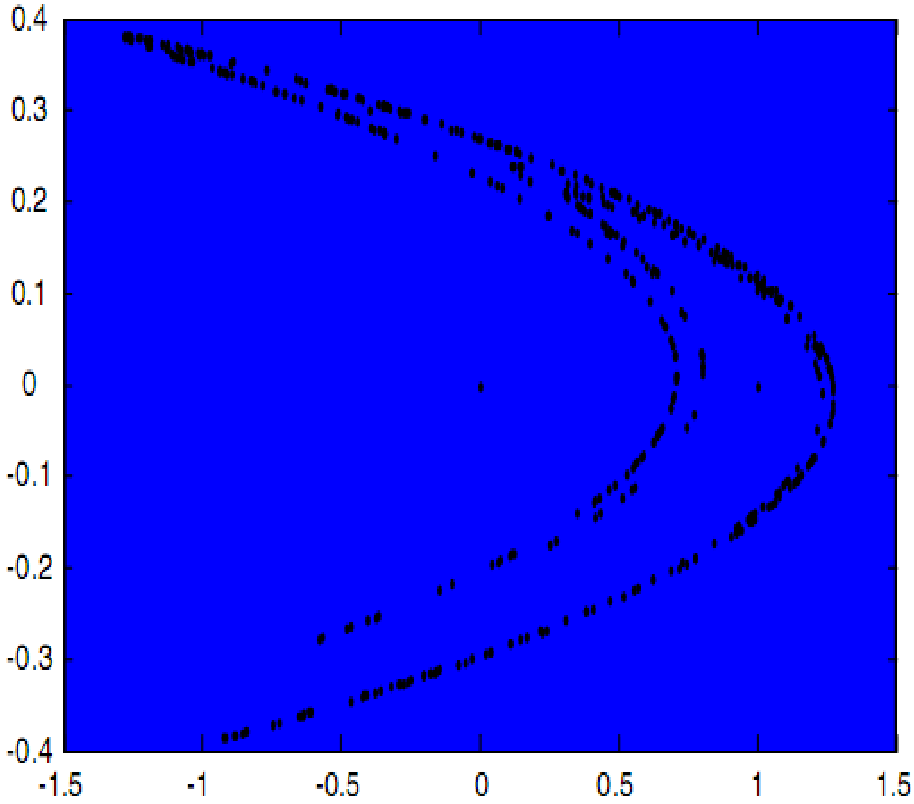


Fig. 7.1: The Hénon attractor.

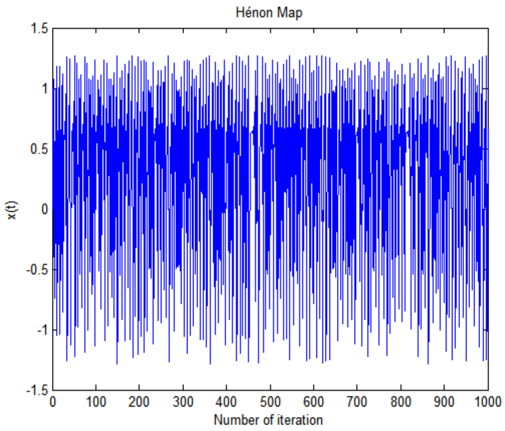


Fig. 7.2: Unpredictability of Hénon map along x-axis.

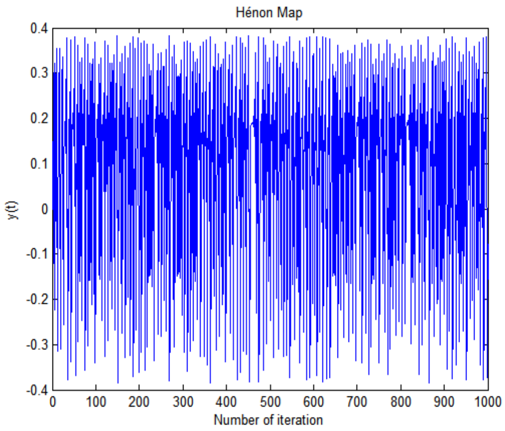


Fig. 7.3: Unpredictability of Hénon map along y-axis.

7.1.2 Mathematical Properties of Hénon Map

1. The Hénon map is composition of three different transformations [141], usually denoted f_1, f_2 and f_3. These conversions are defined below:

$$f_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y+1-\alpha x^2 \end{pmatrix}, \; f_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \beta x \\ y \end{pmatrix} \; \textit{and} \;\; f_3 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix}. \tag{7.2}$$

   From the above definitions, we have $f_{\alpha\beta} = f_3 \circ f_2 \circ f_1$.

2. The Hénon map is one-to-one.

3. The Hénon map is invertible. It is not obvious just from inspection, but it is possible to derive an exact expression for $f_{\alpha\beta}$.

4. For $\beta \neq 0$, the inverse of $f_{\alpha\beta}$ is $f_{\alpha\beta}^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{y}{\beta} \\ -1+\frac{\alpha}{\beta^2}y^2+x \end{pmatrix}.$

The Hénon map has some geometrical properties which inherent stretching and folding in phase space, which offers growth to chaotic actions. The Hénon map is divided into three stages to recognize its correspondence to the stretch and fold action [142]. The following are three phases of Hénon map:

a. Bend up: This property mainly expresses the nonlinear bending in y coordinate given by

$$f_1(x, y) = (x, 1+y-\alpha x^2). \tag{7.3}$$

Along line parallel to $x-$axis ($y = constant$), we have a parabola with the vertex at $(0, 1+y)$.

b. Contraction in x: The second geometrical characteristic is contraction in $x-$direction, which is represented by the following mathematical transformation:

$$f_2(x, y) = (\beta x, 1 + y - \alpha x^2). \tag{7.4}$$

The contraction factor is given by the parameter β, which is 0.3 for the Hénon attractor.

c. Reflection: The reflection along the diagonal is represented by

$$f_3(x, y) = (y, x). \tag{7.5}$$

The effect of the compression is same as apply the unique transformation one time, i.e.,

$$f(x, y) = f_3(f_2(f_1(x, y))). \tag{7.6}$$

A detailed relation of chaos and cryptography are given in Table 3.1.

7.2 Chaos Based Algorithm for S-box Design and Encryption Algorithm

In this section, we have presented the algorithm to synthesize S-boxes that are based on Hénon chaotic map. The algorithm mainly consists of five steps starting from defining initial seed from Hénon chaotic maps to apply permutation of symmetry group to generate S-boxes. This algorithm also demonstrates the application to image encryption systems (see Table 7.1). An instance of the proposed S-box is shown in Table 7.2. This S-box has 16×16 entries obtained from the Hénon chaotic map used in the proposed algorithm.

Table 7.1: Proposed chaos-based algorithm for chaotic S-boxes and image encryption.

S.1: We have taken initial seed of 16×16 distinct values from first component of Hénon chaotic map with properly selected chaotic parameters and initial conditions.

S.2: Convert each of the values in eight bits binary.

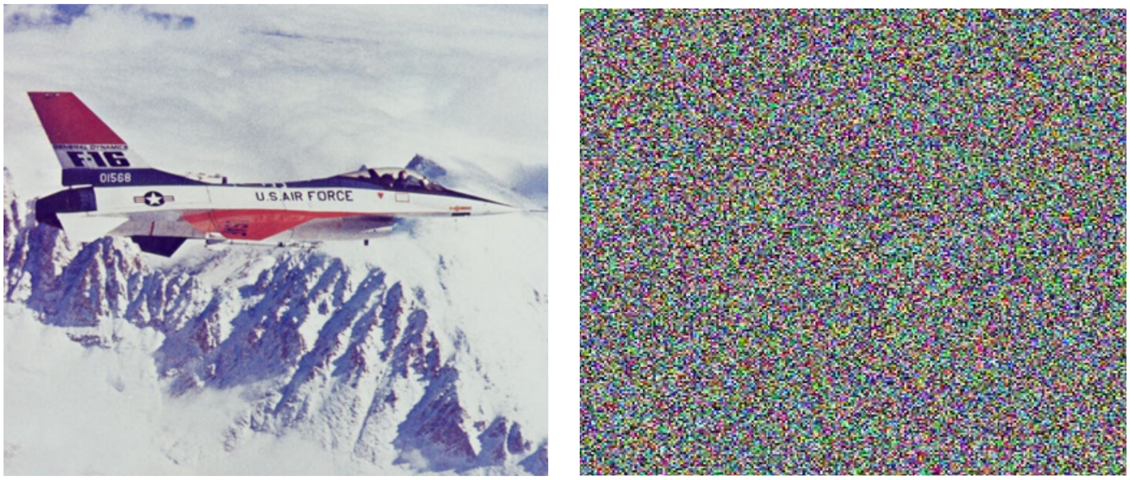
S.3: Apply the permutation of symmetry group of S_8 to each element in step 2.

S.4: Generate the sequence $\mathrm{S}_{n+1} = \lfloor y_{n+1} \times 40320 \rfloor$, we can get the integer sequences that ranges from 0 to 40320.

S.5: The numbers in the sequence produced by in step 4; we select the numbers as the index of the S-boxes order to accomplish the substitution encryption of an image.

Table 7.2: The proposed chaos and permutation symmetry group S_8 S-box.

33	152	0	248	1	174	231	83	43	73	230	185	70	113	58	222
86	17	34	211	21	179	177	187	66	188	254	122	165	197	191	170
146	145	129	30	64	198	6	157	252	173	82	219	132	88	101	162
100	45	19	65	77	208	194	81	192	130	171	32	53	245	37	68
4	62	41	186	212	184	150	79	183	253	94	46	131	16	60	31
36	102	169	205	237	246	144	149	90	154	14	10	24	50	240	72
223	142	117	119	148	23	22	98	178	18	96	118	105	232	155	202
249	103	161	108	199	109	203	128	106	20	47	196	176	244	42	195
49	251	54	163	29	209	48	213	216	110	137	51	217	115	168	236
38	40	15	189	59	135	134	9	39	61	139	234	210	2	180	52
207	243	55	121	8	13	166	175	147	143	3	67	85	172	107	133
226	116	95	153	78	7	228	200	111	63	159	229	126	97	141	26
92	138	112	57	76	218	204	80	125	241	27	220	89	120	167	104
127	28	12	11	255	151	214	93	193	75	239	160	250	56	69	25
201	123	140	71	225	235	233	136	35	158	224	91	242	221	215	247
164	206	5	124	74	181	238	156	227	182	84	114	190	99	44	87



(a) (b)

Fig. 7.4 : Plain (a) and encrypted (b) images of size 256×256.

7.3 Statistical Analyses of Proposed Algorithm

In this section, we mainly discussed the statistical features of an image, which are mainly texture qualities of an image. An image texture is a set of measures considered in image handling designed to measure the observed quality of an image. Image quality offers us evidence about the spatial organization of color or strengths in an image or designated region of an image. The texture features are characteristics, which are used to capture the graphical assets of an image either comprehensively for the complete image or in the neighborhood for sections or stuffs. The graphical appearances of similar areas of real-world images are frequently identified as quality. As an image is made of pixels, texture can be defined as a unit containing jointly connected pixels or cluster of pixels and thus leading to graphical feature of images. An image can be designated with the help of measurements of first order for gray intensities of the pixels inside a locality. The instances of such qualities taken from the image histogram are mean and standard deviation (SD). The characteristics of second order are based on gray level co-occurrence matrix (GLCM) [143]-[147] and it is the best widespread approaches for pixel deviation information. The features of second order are entropy, contrast, homogeneity, energy and correlation of the gray level pixels defined as follows:

$$H = -\sum_i \sum_j p(x_i, x_j) \log p(x_i, x_j), \tag{7.7}$$

$$C = \sum_i \sum_j |i-j|^2 p(i,j), \tag{7.8}$$

$$H_{hom} = \frac{\sum_i \sum_j p(i,j)}{1+|i-j|}, \tag{7.9}$$

$$E = \sum_i \sum_j p(i,j)^2, \tag{7.10}$$

$$r = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j)}{\sigma_i \sigma_j} p(i,j), \tag{7.11}$$

where i and j are two dissimilar gray levels of the image, p is the number of the co-occurrence of gray levels i and j, μ_i, μ_j are mean of i and j levels of image, σ_i and σ_j are the standard deviations at i and j levels of an image. Entropy is used to measure the content of an image with higher value indicating an image with richer details. Contrast is used to measure the intensity change between a pixel and its neighbor over the entire image and is 0 for a constant image. Homogeneity processes the resemblance of gray-scale levels across the image and ranging from zero to unity inclusive. Thus, higher the variations in the grayscale, the higher the GLCM difference and lower the homogeneity. GLCM energy deals with total probability of distinctive grayscale configurations in image, and its value is unity for a constant image. Correlation returns an amount of how interrelated a pixel is to its locality over the entire image, and it is used to measure the joint probability of occurrence of particular pixel sets. The range of correlation coefficient lies between $[-1, 1]$. The encryption through the proposed algorithm is given in Fig. 7.4. Tables 7.3 and 7.4 give comparison of the texture features of original and encrypted images. From the calculated values of entropy (see Table 7.3), we have observed that the entropy values of original images are far from ideal value of entropy, which is eight bits, since data sources are extremely redundant and thus hardly produce evenly scattered random messages, whereas the entropy values of the encrypted images are near to the best value, which means that the suggested encryption procedure is decidedly strong against entropy attacks. The value of contrast for original image is 0.298752, whereas for an encrypted image 7.810361, which clearly reflects that, intensity change between a pixel and its neighbor over the entire encrypted image is high (see Table 7.3). The low value of homogeneity for encrypted image shows the higher GLCM difference and higher differences in the grayscale. Energy analyses of both original and encrypted images reveal that quantity of recurring pairs is low for encrypted image, which is a significance of the proposed algorithm. Finally, we have broken the correlation among the adjacent pixels values as seen from the numerical values of correlation of an encrypted image (see Table 7.4). The values of entropy and contrast of the suggested S-box is greater than Skipjack, Gray and Xyi S-boxes, whereas homogeneity, energy and correlation are smaller than Skipjack, Gray and Xyi S-boxes which clearly reflect the advantage of our proposed chaotic S-boxes (see Table 7.3).

Table 7.3: Comparison of texture features of plain and ciphered images.

Texture features	Plain image	Cipher image			
		Proposed	Skipjack [107]	Gray [107]	Xyi [107]
Entropy	7.431821	7.997300	7.8939	7.9299	7.9127
Contrast	0.298752	7.810361	5.4255	7.7961	7.8942
Homogeneity	0.896043	0.464131	0.5004	0.4567	0.4605
Energy	0.095504	0.028240	0.0232	0.0198	0.0188
Correlation	0.963788	0.009761	0.3123	0.1014	0.1413

Table 7.4: Texture features of plain and ciphered images for color components.

Texture features	Plain image			Cipher image		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.301416	0.29136	0.296596	5.230890	5.423430	5.152970
Homogeneity	0.894101	0.899973	0.897468	0.463997	0.459866	0.465617
Energy	0.112931	0.096232	0.109009	0.026159	0.024618	0.026601
Correlation	0.969755	0.964698	0.956038	0.075238	0.081073	0.075988

7.4 Conclusion

In this work, we have proposed a new procedure for designing chaotic S-boxes and its application in image encryption. This procedure is based on Hénon chaotic map and S_8 permutation. Experimental assessments have been carried out with complete statistical scrutiny, which reveals the strength of the projected procedure against numerous kinds of attacks. Additionally, performance along with valuation and investigations determine that the suggested image encryption algorithm is vastly protected. The proposed encryption scheme is capable of high-speed encryption and decryption, which is appropriate for encryption and broadcast applications.

Chapter 8

A New Implementation of Chaotic S-boxes in CAPTCHA

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a program that can generate and grade tests that humans can pass but current computer programs cannot. The humans can read distorted text as the one shown below, but current computer programs can't. We have proposed a novel chaos based CAPTCHA. We have utilized our proposed S-box and used chaotic iterative relation to select a completely random CAPTCHA for the security of web portal and other user interface multimedia where human interaction with the machine is verified.

8.1 Introduction

A CAPTCHA (an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a kind of test reaction test utilized as a part of figuring to figure out if or not the client is human. The term emerged in 2000 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper of Carnegie Mellon University and John Langford of IBM. The most well-known kind of CAPTCHA was initially designed by Mark D. Lillibridge, Martin Abadi, Krishna Bharat and Andrei Z. Broder. This type of CAPTCHA obliges that the client sort the letters of a twisted picture, in some cases with the expansion of a clouded arrangement of letters or digits that show up on the screen. Since the test is controlled by a machine, rather than the standard Turing test that is directed by a human, a CAPTCHA is once in a while portrayed as a reverse Turing test. This term is equivocal in light of the fact that it could likewise mean a Turing test in which the members are both endeavoring to demonstrate they are the machine [150]-[155].

The CAPTCHA has a few applications for reasonable security. Most bloggers are acquainted with projects that submit fake remarks, normally with the end goal of raising web search tool positions of some site for instance purchasing penny stocks. This is called remark spam. By utilizing a CAPTCHA, no one but people can enter remarks on an online blog. There is no compelling reason to make clients sign up before they enter a remark, and no true blue remarks are ever lost. A few organizations offer free email administrations. Up until a couple of years prior, the majority of these administrations experienced a particular kind of assault: "bots" that would sign up for a great many email accounts consistently. The answer to this issue was to utilize CAPTCHAS to guarantee that just people have acquired free records. In general, free administrations ought to be ensured with a CAPTCHA so as to forestall misuse via mechanized projects. The thought is straightforward: keep a machine from having the capacity to emphasize through the whole space of passwords by obliging it to unravel a CAPTCHA after a specific number of unsuccessful logins.

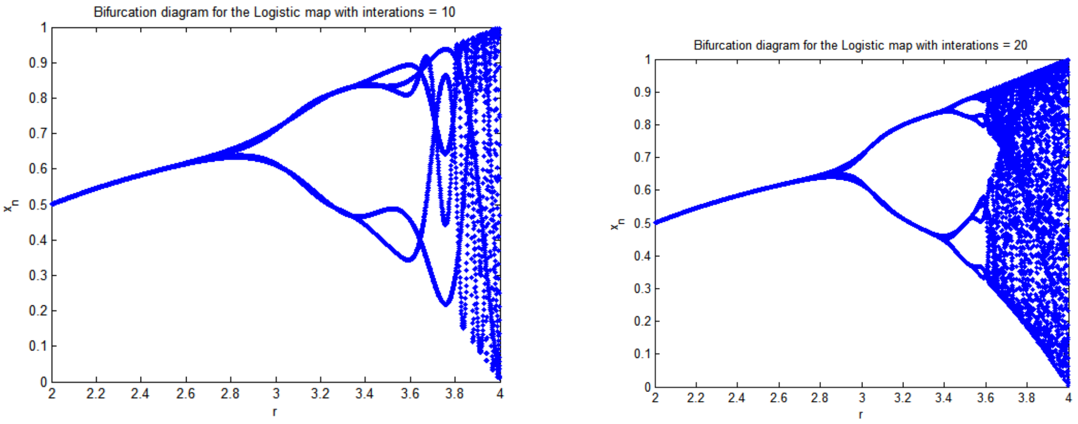
It is off and on again enthralling to keep webpages unlisted to keep others from discovering them effectively. There is a html tag to keep web search tool bots from perusing site pages. The label, on the other hand, doesn't promise that bots won't read a page; it just serves to say "no bots, please." Search motor bots, since they generally have a place with huge organizations, admiration website pages that would prefer not to permit them in. Nonetheless, so as to positively ensure that bots won't enter a site, CAPTCHAs are required. CAPTCHAs additionally offer a conceivable arrangement against email worms and spam. CAPTCHAs must be available. CAPTCHAs built exclusively in light of perusing content, other visual-recognition undertakings, keeps outwardly weakened clients from getting to the secured asset. Any execution of a CAPTCHA ought to permit blind clients to get around the boundary, for instance, by allowing clients to select a sound CAPTCHA. Pictures of content ought to be contorted haphazardly before being displayed to the client. Numerous executions of Captchas use undistorted content, or content with just minor contortions. These usage are defenseless against straightforward computerized assaults. Our main goal in this chapter is to propose an innovative CAPTCHA scheme that is based on chaotic iterative map and S-boxes [156]-[163].

8.2 Logistic Map

The Logistic map is a polynomial mapping (proportionally, repeat connection) of degree 2, regularly referred to as a prototype illustration of how perplexing, chaotic behavior can emerge from extremely basic non-linear dynamical equations. The map was promoted in a seminal 1976 paper by the scientist Robert May [148], to some extent as a discrete-time demographic model closely resembling the Logistic equation initially made by Pierre François Verhulst [149]. Mathematically, the Logistic map is a non-linear repeated connection with a solitary control parameter r

$$x_{n+1} = rx_n(1 - x_n). \tag{8.1}$$

The recurrence relation is started with x_0 between 0 and 1. For $r < 3$, the recurrence relation quickly meets to a limit, i.e. after convergence, each iteration offers the identical value for x. For $3 < r \leq 3.5$, the limiting behavior is an oscillation between two values. Hence two iterations are required before the same x value is obtained. Hence at $r = 3$, a period doubling has occurred. A second period doubling occurs near $r = 3.45$ and another near $r = 3.545$. A careful inspection of the figure below (called a bifurcation diagram) shows that there are more period doublings and that the values r of at which the period doublings occur get closer together (see Fig. 8.1).



(a) (b)

Fig. 8.1: The The chaotic bifurcation diagram of Logistic map.

Near $r = 3.59$, the system becomes chaotic. There is no periodicity in the limiting behavior. Furthermore, for those values of r for which the system is chaotic, the sequence of x_n generated by the logistic map depends sensitively on the beginning value x_0.

8.2.1 Definition

Let $g : I \to I$, has an sensitive dependence on initial conditions at $x \in I$ if $\exists\ \epsilon > 0$ such that $\forall\ \delta > 0$, $\exists\ y$ and m with $|y - x| < \delta$ but $|g^m(y) - g^m(x)| > \epsilon$.

8.2.2 Lyapunov Exponents

The quantitative measures of sensitive dependence on initial condition are calculated through Lyapunov exponents. Let $X \subseteq \mathbb{R}$ and $g : I \to I$ be the function of class C^1 .

8.2.3 Definition

The Lyapunov exponent of g at $x_0 \in X$ is:

$$\lambda_g(x_0) = \lim_{m\to\infty} \frac{1}{m} \ln |(g^m)'(x_0)| = \lim_{m\to\infty} \frac{1}{m} \sum_{j=0}^{m-1} \ln |g'(x_j)| \,, \tag{8.2}$$

where $x_i = g^i(x_0)$. The Lyapunov exponents measure the average values of $\ln|g'|$ along the orbit. The map has sensitive dependence on initial condition at x_0 if $\lambda_g(x_0) > 0$ and doesn't at x_0 if $\lambda_g(x_0) < 0$.

8.3 Chaotic Hyperchaotic Lorenz system

The Lorenz system is inspired by the model or air flow in atmosphere in 1950's and is the first numerical study on chaos. The system dynamics are represented by the following equation:

$$\frac{dx}{dt} = a(y-x), \tag{8.3}$$

$$\frac{dy}{dt} = x(b-z)-y+w, \tag{8.4}$$

$$\frac{dz}{dt} = xy-cz, \tag{8.5}$$

$$\frac{dw}{dt} = -dx. \tag{8.6}$$

The system is hyperchaotic for the parameters are $a = 10$, $b = 28$, $c = 8/3$ and $d = -5$ with the initial conditions $x_0 = y_0 = z_0 = w_0 = 0$ and $-40 \leq x \leq 40,\ -40 \leq y \leq 40,\ -40 \leq z \leq 40,\ -40 \leq w \leq 40$, The system exhibits chaotic behavior for the selected parameters and intervals [113]-[116].

8.3.1 Linear Fractional Transformation S-boxes

In linear fractional transformation (LFT) substitution boxes are constructed by the action $PGL(2, GF(2^8))$ on finite field of order 2^8. The algebraic structure of LFT S-boxes depends on the linear fractional transformations $f(x) = ex + h/kx + l$ where e, h, k and $l \in GF(2^8)$. With this method, one can construct millions of secure S-boxes, particularly the boxes corresponding to affine transformation satisfied the security analysis with optimal value. The construction of LFT S-boxes is as follows :

$$f \quad : \quad PGL(2, GF(2^8)) \times GF(2^8) \rightarrow GF(2^8), \tag{8.7}$$

$$f(x) = \frac{ex+h}{kx+l}. \tag{8.8}$$

The process starts with the action of Galois field on the projective general linear group. The function $f(x)$ depends on the values of e, h, k and $l \in GF(2^8)$, corresponding to every different combination of

e, h, k and l from $GF(2^8)$ one can construct a new 8×8 S-box (see Fig. 8.2).

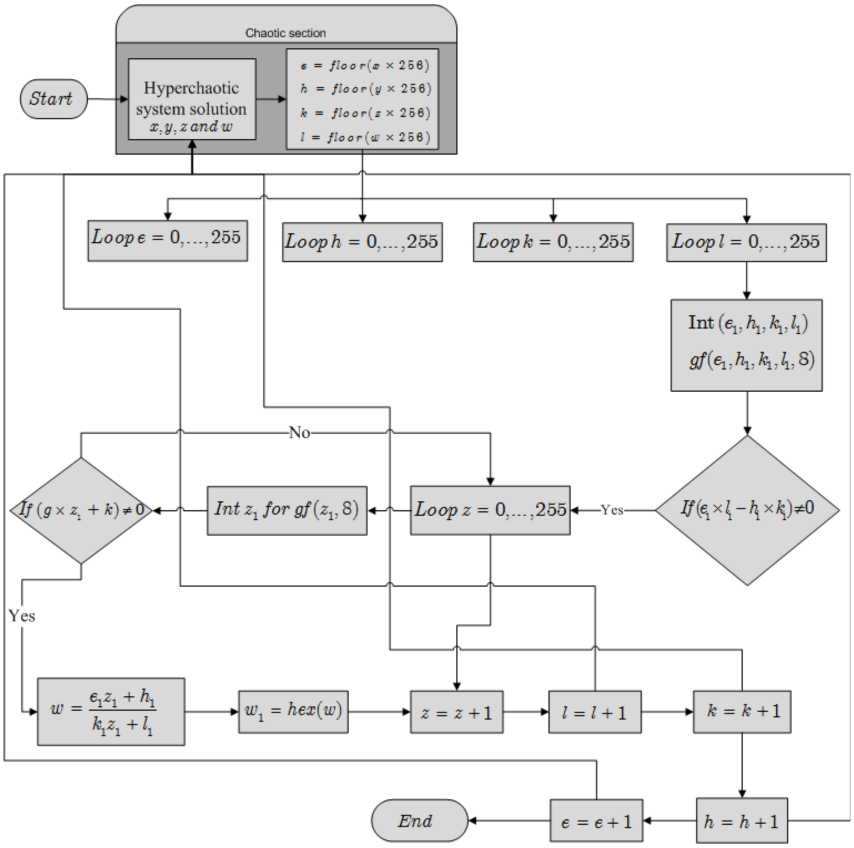


Fig. 8.2: Flow chart for chaotic S-box.

Table 8.1: Proposed chaotic S-box.

243	195	197	44	75	107	12	32	215	234	24	150	6	194	21	83
146	250	58	209	217	206	117	94	161	142	222	2	178	151	96	52
78	55	92	131	81	241	238	47	82	116	86	111	162	158	118	174
232	124	164	64	176	143	122	144	219	184	25	53	130	8	208	31
102	203	225	7	242	60	40	128	72	253	236	216	134	90	120	218
204	182	231	214	153	180	54	85	23	66	1	41	188	156	20	187
9	76	159	171	114	65	106	18	227	179	33	149	223	210	38	69
63	251	113	132	93	154	136	211	104	14	230	228	105	133	88	233
110	244	70	98	108	42	181	167	138	43	221	183	191	50	165	145
152	30	39	68	193	101	5	140	147	246	125	62	84	247	235	67
190	119	89	239	79	201	252	127	207	245	137	0	26	255	73	185
226	249	141	205	155	196	172	91	45	212	202	29	51	17	135	87
4	10	213	170	46	248	199	19	115	77	49	27	103	15	95	198
36	28	254	35	3	48	139	240	186	148	100	168	22	189	166	123
74	34	71	229	13	109	97	200	57	169	192	99	237	175	129	121
112	11	173	157	61	177	59	56	37	160	163	80	126	220	224	16

8.4 Types of CAPTCHA

In this section, we are giving some basic types of CAPTCHA which are used in current email servers (see Fig. 8.3).

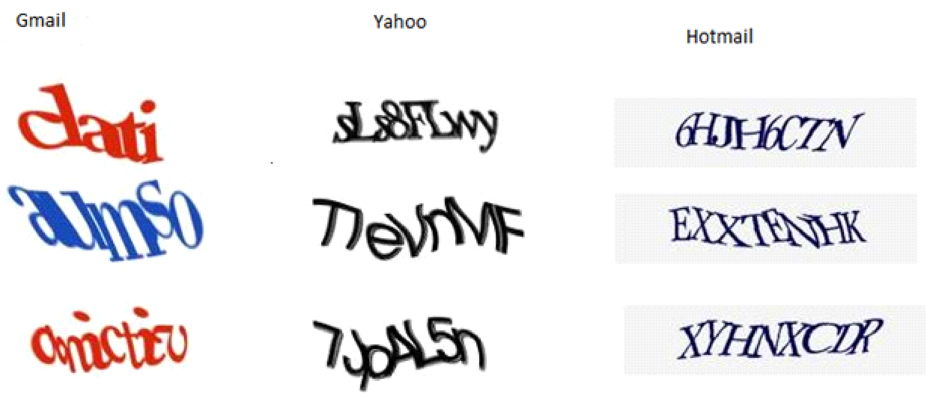


Fig. 8.3: CAPTCHA for Yahoo, Google and MSN.

8.4.1 Types of CAPTCHAs

There are fundamentally three types of CAPTCHAs which are used commonly:

- 1. Visual-based CAPTCHAs

  2. Sound based CAPTCHAs

  3. Graphics based CAPTCHAs

8.4.2 Visual Based CAPTCHAs

Visual based CAPTCHAs come in a few assortments, where the most well-known ones being utilized are contorted content installed as a part of images and shape distinguishment.

Text CAPTCHAs

These are easy to execute. The least complex yet novel methodology is to present the client with a few inquiries which just a human client can comprehend. Samples of such inquiries are:

1. What is thirty minus four?
2. What is the third letter in Mathematics?
3. Which of Green, Friday and Qasim is a color?
4. If yesterday was a Monday, what is the day after tomorrow?

Such inquiries are simple for a human client to illuminate, however, its exceptionally hard to program a machine to tackle them. These are likewise agreeable to individuals with visual incapacity, for example, those with color blindness. Other content Captchas includes content mutilations and the client is asked to distinguish the content covered up. The different executions are Gimpy, EZ-Gimpy (a variation of Gimpy), Pessimal Print, and Baffletext. Gimpy was initially created by Luis von Ahn from Carnegie Mellon University and in addition outlining a rearranged adaptation of Gimpy, called EZ-Gimpy ("Attack," 2002). EZ-Gimpy is right now being utilized by Yahoo! also a comparative form is utilized via Hotmail (Bruno, 2003). The primary distinction in the middle of Gimpy and EZ-Gimpy is that Gimpy has three or more words twisted inside a picture, while EZ-Gimpy generally just has one contorted word in the picture. Likewise, in Gimpy, three or more words must be speculated accurately keeping in mind the end goal to breeze through the test. Both are apparently equivalent in that they both utilize a lexicon that has a sum of 850 words to pick the words that are masked inside the bended picture (Bruno, 2003; Vijayan, 2003). Figs. 8.4 and 8.5, demonstrates a case of Gimpy.

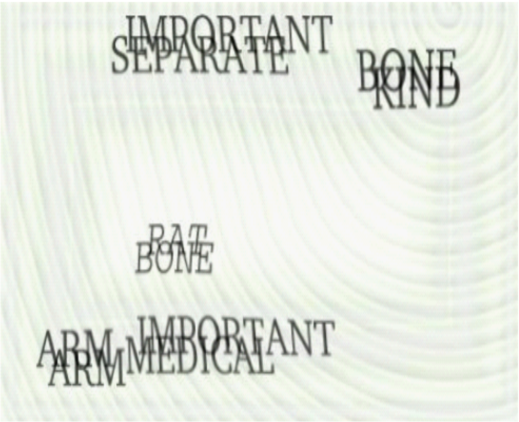


Fig. 8.4: Gimpy CAPTCHA.

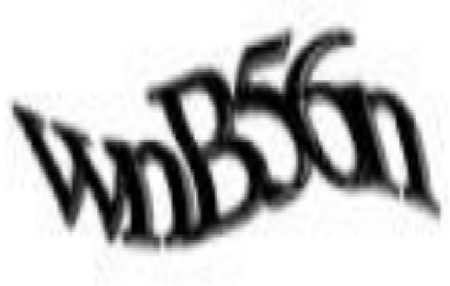


Fig. 8.5: Yahoo's Ez – Gimpy CAPTCHA.

Pessimal Print (see Fig. 8.6) was composed in 2000 via Baird from UC Berkeley/Palo Alto Research Center (PARC) and was one of the first visual based Captchas (Bruno, 2003; Chew & Baird, 2003). The test includes perusing a statement that was corrupted and the test is passed if the saying in the picture was speculated accurately. Notwithstanding, the word reference of the conceivable words that can be picked is just 70 words, which makes it extremely vulnerable to assaults, since it would be not difficult to have a brute-force attack break this CAPTCHA program. Baffletext is the latest visual based CAPTCHA which was produced in 2003 by Monica Chew and Henry Baird from UC Berkeley. It is more effective than the formerly said visual Captchas in that it doesn't utilize words that are found as a part of an English word reference, utilizes numerous distinctive text styles, and does not debase the picture utilizing physical science based corruptions in which other visual Captchas utilization (Chew & Baird, 2003). This is an extraordinary change over EZ-Gimpy and Pessimal Print in that it anticipates brute-force attack and attack from Optical Character Recognition (OCR) programs in that Baffletext gives a complex covering strategy that totally disfigures the picture by embedding squares, rings, and ovals, differing the length and width of the shape, and coloring the state of diverse shades of dark (Chew & Baird, 2003).

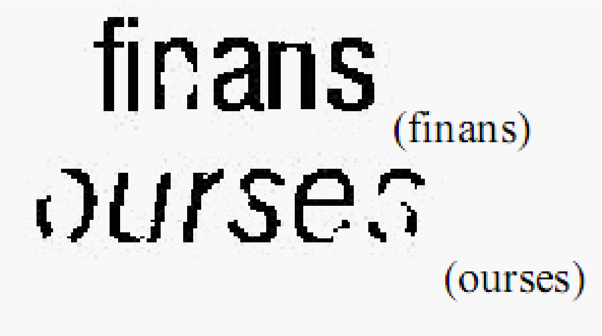


Fig. 8.6: Baffle Text examples.

An illustration of shape distinguishment CAPTCHA is Bongo (see Fig. 8.7). This test exhibits two groups of shapes in which the shapes in each one gathering are identified with one another somehow. An alternate shape is found beneath the two gatherings and the object of this test is to figure out which group the shape fits in with (Ahn, Blum, & Langford, 2004). The test is passed when the group that the shape has a place with is effectively picked. A comparative system which does this is PIX, which uses pictures other than simple shapes. The principle contrast is that it asks what does picture portrayed. Notwithstanding, PIX is not viewed as a CAPTCHA since it can be effortlessly assaulted by an alternate system which can look in its database and discover the picture and the name that is connected with it (Ahn, Blum, & Langford, 2004). This project can be made into a CAPTCHA by distorting the images for the test.

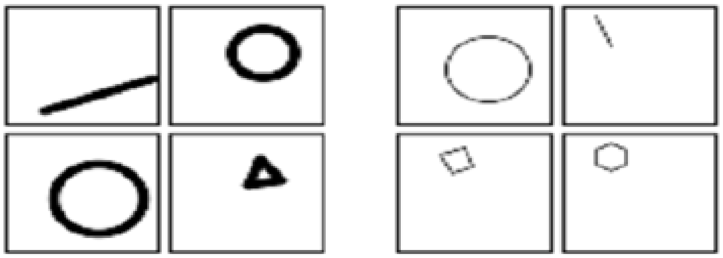


Fig. 8.7: Bongo CAPTCHA.

8.4.3 Sound based

A sound-based CAPTCHA is utilized for the most part to aid the individuals who are hard of hearing or have listened to issues. A sample of a sound-based CAPTCHA is called sounds. This CAPTCHA is utilized as a part of Hotmail, Yahoo!, and Altavista notwithstanding the visual-based Captchas when enlisting for a record for each of these email administrations. The test plays an audio clip which contains the recording of a distorted word or grouping of numbers and it is passed if the saying or numbers are speculated effectively (Robinson, 2002).

8.4.4 Graphic CAPTCHAs

Graphic Captchas are difficulties that include pictures or substances that have a comparability that the clients need to figure. They are visual riddles, like Mensa tests. The machine creates the riddles and grades the answers, yet is itself not able to solve it. CAPTCHA that obliges two steps to be passed. To begin with step guest clicks somewhere else on the picture that made out of a couple of pictures and chooses thusly a solitary picture. Second step the chose picture is stacked. It is developed, however abundantly contorted. Likewise variations of the answer are stacked on the customer side. The guest ought to choose a right reply from the set of the proposed words.

8.5 Proposed Chaotic CAPTCHA Based on S-box

In the present section, we have presented a proposed CAPTCHA technique that is based on chaotic Logistic map and projective S-box. The algorithm of proposed CAPTCHA technique is given as follows:

Algorithm

1. *Firstly, we have constructed a chaotic projective S-box that is based on linear fractional transformation and hyperchaotic Lorenz system.*

2. *Secondly, we have taken logistic map for the selection of random numbers from our proposed chaotic S-box.*

3. *Thirdly, we have generated a random string of standard implementation-independent characters (ASCII codes : 32-127) from our projected S-box with the help of Logistic map.*

4. *Fourthly, we have limited all these standard characters within the limit of keyboard characters in order to use our CAPTCHA effectively.*

5. *Fifthly, We have generated M CAPTCHA questions in a pool related to the string generated in step 4. What is a color of a specific character in a string of length N?).*

6. *Lastly, after answering successfully, the user will be able to connect to the server being a human.*

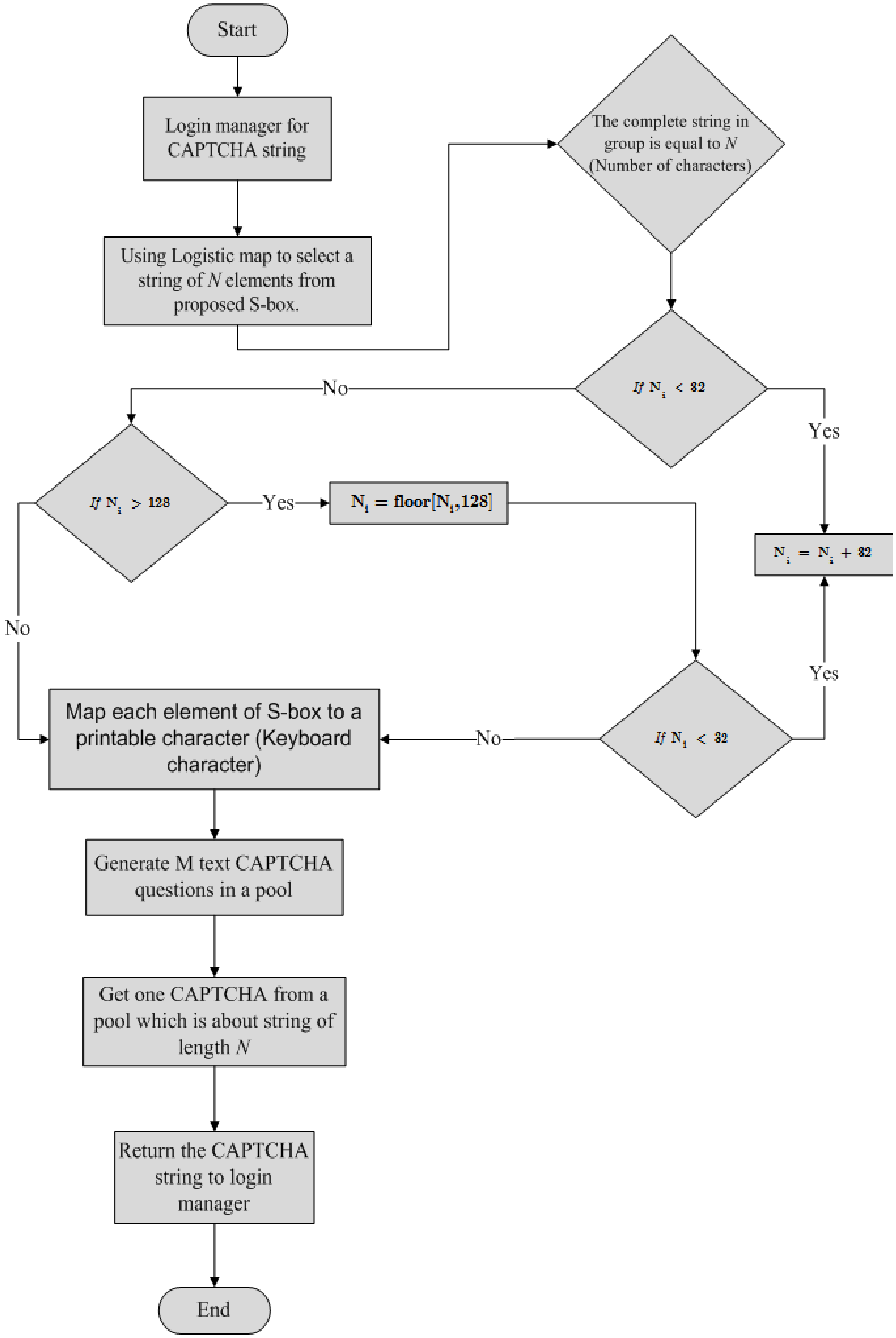
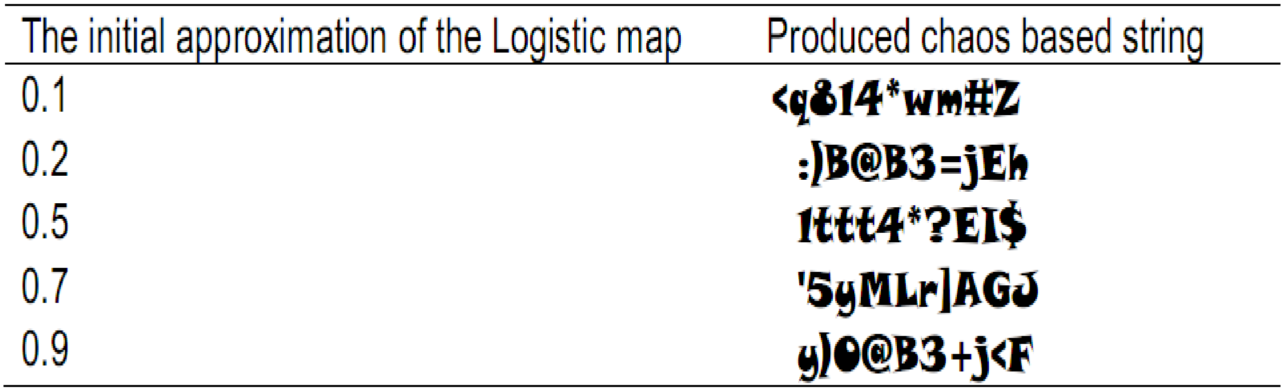

Start
Login manager for CAPTCHA string
Using Logistic map to select a string of N elements from proposed S-box.
The complete string in group is equal to N (Number of characters)
If N_i < 32
No
Yes
If N_i > 128
Yes
N_i = floor[N_i,128]
N_i = N_i + 32
No
Yes
Map each element of S-box to a printable character (Keyboard character)
No
If N_i < 32
Generate M text CAPTCHA questions in a pool
Get one CAPTCHA from a pool which is about string of length N
Return the CAPTCHA string to login manager
End

Fig. 8.8: Flow chart for proposed chaotic CAPTCHA.

Table 8.2: Output of proposed chaos based string.



The initial approximation of the Logistic map	Produced chaos based string
0.1	<q814*wm#Z
0.2	:)B@B3=jEh
0.5	1ttt4*?EI$
0.7	'5yMLr]AGJ
0.9	y)O@B3+j<F

The outputs produced with the help of anticipated technique clearly elucidate sensitivity to initial conditions. We are then select some questions from our generated pool about the chaotic string. Finally, we have answered selected questions in constructed pool and login to web portal as a human not as a machine. Our proposed schemes of CAPTCHA generation, qualify for good candidates of secure CAPTCHA. Moreover, CAPTCHA generated through suggested chaotic technique is highly sensitive to chaotic parameters and initial conditions which create confusion and diffusion capability in a single step which is necessary for any secure communication mechanism.

8.6 Conclusion

To conclude, we have presented an approach that relates to the fields of chaos and pattern recognition to cryptography. We have not only constructed chaotic S-box, but also used it in a CAPTCHA implementation. This methodology permits us to practice the evolution of a chaotic system near a phase transition to embed and protect a secret token that can subsequently be used for cryptographic purposes. Our method can be enthusiastic and candidly executed on a widespread diversity of prevailing computer structures and devices and, in our opinion, offers a momentous stage advancing in the security of confidential data as compared to the presently existing techniques. We are confident that our outcomes can open a region for future research. The proposed idea is tested for selecting different initial conditions and chaotic parameters to generate substantial structures in varieties potentially acceptable to human users. Potential future directions include sound and graphics CAPTCHAs applications.

Chapter 9

A Copyright Protection Using Watermarking Scheme Based on Nonlinear Permutation and its Quality Metrics

The advancement of the Internet stretched as frequently as it can in the openness of modernized data, for instance, image, audio and video for public usage. Digital watermarking is a building that delivers surety and support to data check, security and copyright protection of cutting edge media. This chapter combines the purpose of investment study watermarking definition, thought and the guideline responsibilities in this field, for instance, orders of watermarking process that advice which watermarking procedure should be used. It starts with defining some basics and proposing scheme which is based on nonlinear permutation, Least Significant Bits (LSB), chaotic Logistic and Gauss maps. Finally distance metrics for proposed scheme discussed to assure the watermark in test image.

9.1 Digital Watermarking Technology

As a rising engineering, digital watermarking includes the plans and hypotheses of diverse subject scope such as indicator transforming, cryptography, likelihood hypothesis and stochastic hypothesis, system innovation, algorithms configuration and different methods [165]. Advanced watermarking shrouds the copyright data into the computerized information through certain algorithm. The secret data is to be implanted might be some content, serial number, organization logo and pictures with some exceptional essentialness. This secret data is connected to the advanced information (images, audio and videos) to guarantee the security, information confirmation, distinguishing proof of holder and copyright insurance. The watermark could be covered up in the advanced information either noticeably or imperceptibly. For a solid watermark installing, a great watermarking strategy is required to be connected. Watermark could be inserted either in spatial or frequency domain. Both the domains are distinctive and have their own particular advantages and disadvantages and are utilized as a part of diverse situation.

9.2 Preliminaries

In this section, we defined some basic definitions that will be used in subsequent sections.

9.2.1 Distance Function

A distance function on a given set M is a function $d : M \times M \to R$, where R denotes the set of real numbers, that satisfies the following conditions:

1. $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$. (Distance is positive between two different points, and is zero precisely from a point to itself.)
2. It is symmetric: $d(x, y) = d(y, x)$. (The distance between x and y is the same in either direction.)
3. It satisfies the triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$. (The distance between two points is the shortest distance along any path).

Such a distance function is known as a metric. Together with the set, it makes a metric space.

9.2.2 Euclidean Distance of Images

Among all the image metrics, Euclidean distance is the most commonly used due to its simplicity. Let x and y be two $M \times N$ images, $x = \left(x^1, x^2, x^3, ..., x^{MN}\right)$, $y = \left(y^1, y^2, y^3, ..., y^{MN}\right)$, where x^{kM+l} , y^{kN+l} are the gray level at location (k, l). The Euclidean distance is given by:

$$d_E(x, y) = \sqrt{\sum_{k=1}^{MN} (x^k - y^k)^2}. \tag{9.1}$$

All the $M \times N$ images are easily discussed in $M \times N$ dimensional Euclidean space, called image space. It is natural to adopt the bases $\alpha = (\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{MN})$ to form a coordinate system of the image space where α_{kN+l} corresponds to an ideal point source with unit intensity at the location (k, l). Thus an image $x = \left(x^1, x^2, x^3, ..., x^{MN}\right)$, where x^{kN+l}, is the gray level at the $(k, l)^{th}$ pixel, is represented as a point in the image space, and x^{kN+l} is the coordinate with respect to α_{kN+l}. The origin of the image space is an image whose gray levels are zero everywhere [171].

9.2.3 Squared Euclidean Distance

The standard Euclidean distance can be squared in order to place progressively greater weight on objects that are farther apart. In this case, the equation becomes

$$d_{SE}^2(x,y) = \sum_{k=1}^{MN} (x^k - y^k)^2, \tag{9.2}$$

The squared Euclidean distance is not a metric as it does not satisfy the triangle inequality, however it is frequently used in optimization problems in which distances only have to be compared. It is also referred to as quadrance within the field of rational trigonometry.

9.2.4 Mahalanobis Distance

To account for differences in variance between the variables, and to account for correlations between variables, we use the Mahalanobis distance:

$$d_{MD}^2(x,y) = (x_i - y_i)\mathbf{cov}^{-1}(x_i - y_i)^T, \tag{9.3}$$

where $\mathbf{cov}^{-1}$ is covariance of random variables x and y respectively.

9.2.5 Normalized Euclidean Distance

If the covariance matrix is the identity matrix, the Mahalanob is distance reduces to the Euclidean distance. If the covariance matrix is diagonal, then the resulting distance measure is called a normalized Euclidean distance [28]:

$$d(\overrightarrow{x}, \overrightarrow{y}) = \sqrt{\sum_{i=0}^{N} \frac{(x_i - y_i)^2}{s_i^2}}, \tag{9.4}$$

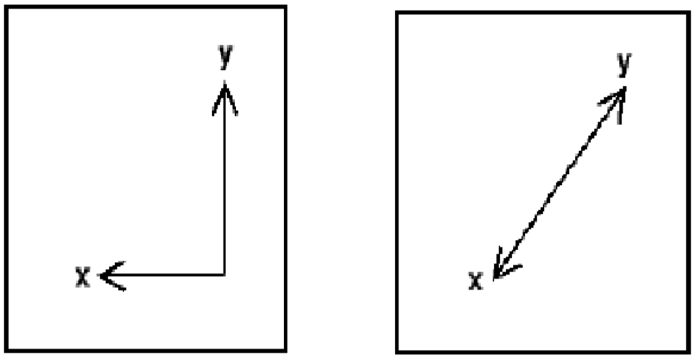
where s_i is the standard deviation of the x_i and y_i over the sample set.

9.2.6 Manhattan Distance

The distance between two points in a grid based on a strictly horizontal and/or vertical path (that is, along the grid lines), as opposed to the diagonal or "as the crow flies" distance. The Manhattan distance or city block distance is the simple sum of the horizontal and vertical components, whereas the diagonal distance might be computed by applying the Pythagorean theorem. The mathematical expression for the city block distance is:

$$d_M(x,y) = \sum_{k=1}^{MN} \left| x^k - y^k \right|. \tag{9.5}$$

The Manhattan distance function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components. The City block distance is always greater than or equal to zero. The measurement would be zero for identical points and high for points that show little similarity. The following figure illustrates the difference between Manhattan and Euclidean distances:



(a) (b)

Fig. 9.1: Difference between Manhattan and Euclidean distances.

9.2.7 Cosine Distance Measure

Hearst (1997) used the ad hoc Cosine-distance measure for scoring the text blocks in the TextTiling algorithm. The concept of the Cosine-distance measure is explained in [173]. Given two N dimensional vectors x and y then the cosine similarities between them are calculated as follows [173]:

$$d_{CS}(x, y) = \frac{\sum_{i=0}^{N} x_i . y_i}{\sqrt{\left(\sum_{i=0}^{N} x_i^2\right)\left(\sum_{i=0}^{N} y_i^2\right)}}. \tag{9.6}$$

The cosine distance is defined as follows:

$$d_{CD}(x, y) = 1 - \frac{\sum_{i=0}^{N} x_i . y_i}{\sqrt{\left(\sum_{i=0}^{N} x_i^2\right)\left(\sum_{i=0}^{N} y_i^2\right)}}. \tag{9.7}$$

This measure also yields a score ranging from 0 to 1; unlike the original Cosine-distance measure, higher scores indicate higher possibility of topic shift [172].

9.2.8 Distance Correlation

In statistics and in probability theory, distance correlation is a measure of statistical dependence between two random variables or two random vectors of arbitrary, not necessarily equal dimension. An important property is that this measure of dependence is zero if and only if the random variables are statistically independent. This measure is derived from a number of other quantities that are used in its specification, specifically: distance variance, distance standard deviation and distance covariance. The correlation between two data points $x = \left(x^1, x^2, x^3, ..., x^{MN}\right)$ and $y = \left(y^1, y^2, y^3, ..., y^{MN}\right)$ in R^N and is given by [172]:

$$Corr(x,y) = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \overline{y})^2}}, \tag{9.8}$$

which can also be represented by:

$$Corr(x,y) = \frac{(x_i - \overline{x})(y_i - \overline{y})}{||(x_i - \overline{x})||\,||(y_i - \overline{y})||}, \tag{9.9}$$

where $x.y$ is the scalar product of x and y. The correlation based distance is given as follows:

$$d_{DC}(x,y) = \sqrt{1 - Corr(x,y)}, \tag{9.10}$$

where $x, y \in R^N$. Note that the bound for the distance correlation is $0 \leq d_{DC}(x,y) \leq 1$. The following are metric properties of correlation distance:

$$\begin{aligned} 1.\ d_{DC}(x,x) &= 0, & (9.11)\\ 2.\ d_{DC}(x,y) &= d_{DC}(y,x), & (9.12)\\ 3.\ d_{DC}(x,z) &\leq d_{DC}(x,y) + d_{DC}(y,z), & (9.13) \end{aligned}$$

Some statistical properties of distance correlations are:

1. $d_{DC}(x,y) = 0 \Leftrightarrow$ x and y are independent.
2. $d_{DC}(x,y) = 1$, implies that the dimensions of the linear spaces spanned by x and y samples, respectively, are almost surely equal and if we assume that these subspaces are equal, then in this subspace $y = A_1 + a_2 A_2 x$, for some vector A_1, scalars a_2 y and orthonormal matrix A_3.

9.2.9 Mutual Information

The first goal of a prediction model is to minimize the uncertainty on the dependent variable. A good formalization of the uncertainty of a random variable is given by Shannon and Weaver's [174]. While first developed for binary variables, it has been extended to continuous variables. Let X and Y be two random variables (they can have real or vector values). We denote $\mu_{X,Y}$ the joint probability density function (pdf) of X and Y. We recall that the marginal density functions are given by

$$\mu_X(x) = \sum_y \mu_{X,Y}(x,y), \tag{9.14}$$

and

$$\mu_Y(y) = \sum_x \mu_{X,Y}(x, y). \tag{9.15}$$

Let us now recall some elements of information theory. The uncertainty on Y is given by its entropy defined as

$$H(Y) = -\sum_y \mu_Y(y) \log \mu_Y(y). \tag{9.16}$$

If we get knowledge on Y indirectly by knowing X, the resulting uncertainty on Y knowing X is given by its conditional entropy, that is

$$H\left(Y|X\right) = -\sum_{x,y} \mu_{X,Y}(x, y) \log \mu_{X,Y}(y|x). \tag{9.17}$$

The joint uncertainty of the (X, Y) pair is given by the joint entropy, defined as

$$H(X, Y) = -\sum_x \sum_y \mu_{X,Y}(x, y) \log \mu_{X,Y}(x, y). \tag{9.18}$$

The mutual information between X and Y can be considered as a measure of the amount of knowledge on Y provided by X (or conversely on the amount of knowledge on X provided by Y). Therefore, it can be defined as [175]:

$$I(X, Y) = H(Y) - H(Y|X), \tag{9.19}$$

which is exactly the reduction of the uncertainty of Y when X is known. If Y is the dependent variable in a prediction context, the mutual information is thus particularly suited to measure the pertinence of X in a model for Y [175]. Using the properties of the entropy, the mutual information can be rewritten into

$$I(X, Y) = H(X) + H(Y) - H(X, Y), \tag{9.20}$$

that is, according to the previously recalled definitions, into [175]:

$$I(X, Y) = \sum_{x,y} \mu_{X,Y}(x, y) \log \frac{\mu_{X,Y}(x, y)}{\mu_X(x)\mu_Y(y)}. \tag{9.21}$$

Therefore we only need to estimate in order to estimate the mutual information between X and Y by Eqs. (9.19) – (9.21).

9.2.10 Mutual Information Variation

The joint entropy minus mutual information

$$d(X, Y) = H(X, Y) - I(X, Y), \tag{9.22}$$

where $H(X,Y)$ is joint entropy of random variables X and Y and $I(X,Y)$ is the joint mutual information of random variables X and Y respectively [175].

9.2.11 Normalized Mutual Information Variation

The mutual information variation divided by joint entropy. The mathematical expression for normalized mutual information variation is given as follows [175]:

$$d_N(X,Y) = \frac{I(X,Y)}{H(X,Y)}. \tag{9.23}$$

9.3 Some Basic Properties of Boolean Functions

9.3.1 Boolean Functions

A boolean function with m inputs is a mapping $g : GF(2)^m \to GF(2)$, where $m \in \mathbb{N}$. The Boolean function $g : GF(2)^m \to GF(2)$ is a affine one when it can be represented as $g(x) = a_m x_m \oplus a_{m-1}x_{m-1} \oplus ... \oplus a_1 x_1 \oplus a_0$ where $x = [x_m, x_{m-1}, ..., x_1]$ and $a_i \in GF(2)$, $i = 0, 1, 2, .., m$. The affine function g is linear when $a_0 = 0$. Let α_i be n dimensional binary vector being the binary representation of an integer which can be written in decimal form, i.e., $\alpha_0 = [0, 0, 0, ..., 0]$, $\alpha_1 = [0, 0, 0, ..., 1], ..., \alpha_{2^m-1} = [1, 1, 1, ..., 1]$. Then the binary vector $[g(\alpha_0), g(\alpha_1), ..., g(\alpha_{2^m-1})]$ is called the truth table of the Boolean function $g : GF(2)^m \to GF(2)$. The truth table uniquely describes the Boolean function, hence writing we mean usually the binary vector representing its truth table. For a given Boolean function we define the polar function $\widehat{g}(x) = (-1)^{g(x)}$ which takes the values from the set $\{-1, 1\}$. We denote $\mathbf{wt}(a)$ the Hamming weight of the binary vector $a = [a_m, a_{m-1}, ..., a_1] \in GF(2)^m$, which is the number of ones in a, i.e. $\mathbf{wt}(a) = \sum_{i=1}^{2^m-1} a_i$. For two vectors $a, b \in GF(2)^m$ their Hamming distance is defined as the number of places where the coordinates of these vectors are different, i.e., $d(a, b) = \mathbf{wt}(a \oplus b)$. For given two Boolean functions $g, h : GF(2)^m \to GF(2)$, their Hamming distance is defined as the number of places at which are different their truth tables, i.e., $d(g, h) = \#\{x \in GF(2)^m | g(x) \neq h(x)\} = \mathbf{wt}(g \oplus h) = \sum_{x \in GF(2)^m} g(x) \oplus h(x)$, where $\mathbf{wt}(g \oplus h)$ is the Hamming weight of the function $g \oplus h$ [167].

Definition 96 *For a Boolean function f, the Walsh Hadamard transform is defined by*

$$\widehat{\Omega}_g(\alpha) = \sum_{x \in B^n} \widehat{g}(x)\widehat{L}_\alpha(x), \tag{9.24}$$

where $\widehat{g}(x) = (-1)^{g(x)}$ and B^n is the set of Boolean function. We denote the maximum absolute value taken by the Walsh Hadamard transform by

$$WHT_{\max}(g) = \max_{\alpha \in B^n} |\widehat{\Omega}_g(\alpha)|. \tag{9.25}$$

Definition 97 *The nonlinearity N_g of a Boolean function f is its minimum distance to any affine function. It is given by*

$$N_g = \frac{1}{2}(2^n - WHT_{\max}(g)). \tag{9.26}$$

It is known that for n even, the maximum non-linearity attainable is [164]:

$$N_{\max}(g) = 2^{n-1} - 2^{\frac{n}{2}-1}, \tag{9.27}$$

but such functions (bent functions) are not balanced. For a given permutation $g \in P_n$ is the set of all permutations over $GF(2^n)$ we define its nonlinearity as

$$N_g = \min_i \left(N_{g_i}, N_{g_i^{-1}} \right),$$

where $g_i = (g_1, g_2, ..., g_n)$ and $g_i^{-1} = (g_1^{-1}, g_2^{-1}, ..., g_n^{-1})$ are coordinates of the original and the inverse permutations respectively.

9.3.2 Permutation Polynomials

Let p be a prime and q is a power of p. Let F_q be a finite field with elements q. A polynomial $g \in F_q[x]$ is called a permutation polynomial of F_q if the mapping $x \mapsto g(x)$ is a permutations of F_q. Every function from F_q to F_q can be represented by a polynomial in $F_q[x]$. In fact, if $\psi : F_q \to F_q$ is an arbitrary function from F_q to F_q, then there exist a unique polynomial $g \in F_q[x]$ with $\deg(g) \leq q-1$ representing ψ, that is $g(d) = \psi(d)$ for all $d \in F_q$. The polynomial g can be found by Lagrange's interpolation technique for the function ψ. If ψ is already given as a polynomial function, say $c \mapsto f(c)$ where $f \in F_q[x]$, then g can be obtained from by reduction modulo $x^q - x$. We call permutations of F_q *PPs* over F_q. These permutation polynomials play a central role in both arithmetic and combinatorial aspects of finite fields. The permutation polynomials have important applications in Coding theory, Cryptography, Finite Geometry, Combinatorics and Computer science among other fields [168]-[170].

Theorem 98 *Given a permutation f, its nonlinearity can be calculated as*

$$N_g = \min_{\beta \in L_n^n} \min_{i=1,2,...,n} (N_{(g*\beta)_i}) = \min_{\beta \in L_n^n} \min_{i=1,2,...,n} (N_{(f^{g^{-1}}*\beta)_i}), \tag{9.28}$$

where L_n^n is the set of all linear permutations and $(g * \beta)_i$ represents the ith coordinates of a composite permutations $g * \beta$.

9.3.3 Exponent Permutations

The exponent permutations is of the form

$$f(x) = (h(x))^e \mod p(x), \tag{9.29}$$

where $p(x)$ is irreducible polynomial of degree n which generates a Galois field $GF(2^n)$, e is any positive integer and $h(x) \in GF(2^n)$ [164].

9.3.4 Example

Let us consider $GF(2^3)$ generated by $p(x) = x^3 + x^2 + 1$ and two permutations given as follows:

$$f_1(x) = (h(x))^2 \mod p(x), \; f_2(x) = (h(x))^3 \mod p(x). \tag{9.30}$$

The permutations along the Walsh transforms are given below (see Tables 9.1 – 9.2):

Table 9.1: Permutation and Walsh transform for f_1.

$GF(2^3)$	f_1	F_1	F_2	F_3	F
0	0	4	4	4	28
1	1	-4	0	0	-4
2	4	0	0	0	0
3	5	0	0	0	0
4	7	0	-4	0	-8
5	6	0	0	0	0
6	3	0	0	-4	-16
7	2	0	0	0	0

Table 9.2: Permutation and Walsh transform for f_2.

$GF(2^3)$	$f\ _2$	F_1	F_2	F_3	F
0	0	4	4	4	28
1	1	2	0	0	2
2	5	0	-2	0	-4
3	2	-2	-2	0	-6
4	6	0	-2	-2	-12
5	4	-2	2	-2	-6
6	7	0	0	-2	-8
7	3	-2	0	2	6

In space $GF(2^4)$, we have seen that the set of linear exponents $\{1, 2, 3, 4, 8\}$ and the set of nonlinear ones $\{7, 11, 13, 14\}$. The rest of exponents do not give permutations they are factor of $2^4 - 1 = 15$ and they do not have inverses. Nonlinear exponent permutations have the same maximum nonlinearity equal to 4 [164].

Corollary 99 *Any permutation*

$$(h(x))^e \mod p(x), \tag{9.31}$$

where p(x) is a generator of GF(2^n), is a linear permutation for e=1,2,4,...,2^{n-1}.

Corollary 100 *(h(x))^e permutes $GF(q)$ if and only if (e,q-1)=1.*

9.4 Algebraic Preliminaries and S-box Construction

For the study of algebraic construction of the S-box a theorem is stated here without proof [166].

Theorem 101 *Let p be a non-zero element of a principle ideal domain R then R/(p) will be a field if and only if p is irreducible. According to this theorem, for a prime p ,Galois field GF(p^n) is constructed by using a generating polynomial q(x) of degree n taking*

$$GF(p^n) = \frac{GF(2)[x]}{< q(x) >}. \tag{9.32}$$

In AES algorithm, the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ is used to generate underlying field $GF(2^8)$. All bytes b in Rijndeal are interpreted as elements of this field represented by a polynomial $c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5 + c_6x^6 + c_7x^7$, where each bit $c_i \in GF(2)$ and $b \in GF(2^8)$. In this field, addition $\oplus$ and multiplication $\otimes$ are defined by the XOR operation and polynomial multiplication modulo the generating polynomial respectively. A S-box is a transformation $S : GF(p^n) \to GF(p^n)$. In AES the S-box $S : GF(2^8) \to GF(2^8)$ is constructed by substitution each element with its inverse and applying a suitable affine transformation $S : x \to Ax^{-1} \oplus b$, where A is belonging from general linear groups of degree 8 over $GF(2)$ and $b \in GF(2^8)$.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{9.33}$$

The motivation for this S-box design is to be resistant to differential, linear cryptanalysis and interpolation attacks. The core design is a simple transformation this mapping has a simple algebraic expression. However, the simplicity itself makes it vulnerable to attacks like the interpolation attack. The purpose of the constant translation vector b is to ensure that there exists no fixed and conjugate fixed points (i.e. $\nexists\ x \in GF(2^8)$ such that $S(x) = x$ and $S(x) = \overline{x}$) in the S-box.

9.4.1 Design of Copyright Protection Technique

The proposed watermark technique is based on small field of sixteen elements $GF(2^4)$ i-e; whose elements have of the form:

$$GF(2^4) = \frac{\mathbb{Z}_2[X]}{< p(x) >} = \left\{ b_0 + b_1 x + b_2 x^2 + b_3 x^3 :\ b_i \in \mathbb{Z}_2 \right\}, \tag{9.34}$$

where $p(x) = x^4 + x + 1$ is an irreducible polynomial. The following tables shows the elements of $GF(2^4)$ and nonlinear permutations based on $GF(2^4)$ (see Table 9.3-9.5):

Table 9.3: Representations of Galois field $GF(2^4)$.

Exp.	Polynomials	Binary	Decimal
$-\infty$	0	0000	0
α^0	1	0001	1
α^1	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha + 1$	0011	3
α^5	$\alpha^2 + \alpha$	0110	6
α^6	$\alpha^3 + \alpha^2$	1100	12
α^7	$\alpha^3 + \alpha + 1$	1011	11
α^8	$\alpha^2 + 1$	0101	5
α^9	$\alpha^3 + \alpha$	1010	10
α^{10}	$\alpha^2 + \alpha + 1$	0111	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{14}	$\alpha^3 + 1$	1001	9

Table 9.4: Nonlinear permutation generated with Galois field $GF(2^4)$.

$h(x)$	Nonlinear permutations of $GF(2^4)$.			
	$f(x) = (h(x))^7 \bmod p(x)$		$f(x) = (h(x))^{11} \bmod p(x)$	
	Polynomials	Decimals	Polynomials	Decimals
0	0	0	0	0
1	1	1	1	1
x	x^3+x+1	11	x^3+x^2+x	14
x^2	x^3+1	9	x^3+x+1	11
x^3	x^3+x^2	12	x^3	8
$x+1$	x^3+x^2+1	13	x^3+1	9
x^2+x	x^2+x	6	x^2+x+1	7
x^3+x^2	x^3+x^2+x+1	15	x^3+x^2	12
x^3+x+1	$x+1$	3	x^2	4
x^2+1	x^3+x^2+x	14	x^3+x^2+1	13
x^3+x	x^3	8	x^3+x	10
x^2+x+1	x^2+x+1	7	x^2+x	6
x^3+x^2+x	x^2	4	x	2
x^3+x^2+x+1	x^3+x	10	x^3+x^2+x+1	15
x^3+x^2+1	x	2	x^2+1	5
x^3+1	x^2+1	5	$x+1$	3

Table 9.5: Nonlinear permutation generated with Galois field $GF(2^4)$.

$h(x)$	Nonlinear permutations of $GF(2^4)$.			
	$f(x) = (h(x))^{13} \bmod\ p(x)$		$f(x) = (h(x))^{14} \bmod\ p(x)$	
	Polynomials	Decimals	Polynomials	Decimals
0	0	0	0	0
1	1	1	1	1
x	x^3+x^2+1	13	x^3+1	9
x^2	x^3+x^2+x	14	x^3+x^2+1	13
x^3	x^3+x	10	x^3+x^2+x+1	15
$x+1$	x^3+x+1	11	x^3+x^2+x	14
x^2+x	x^2+x	6	x^2+x+1	7
x^3+x^2	x^3	8	x^3+x	10
x^3+x+1	x	2	x^2+1	5
x^2+1	x^3+1	9	x^3+x+1	11
x^3+x	x^3+x^2+x+1	15	x^3+x^2	12
x^2+x+1	x^2+x+1	7	x^2+x	6
x^3+x^2+x	x^2+1	5	$x+1$	3
x^3+x^2+x+1	x^3+x^2	12	x^3	8
x^3+x^2+1	$x+1$	3	x^2	4
x^3+1	x^2	4	x	2

The S-box is generated by determining the multiplicative inverse for a given nonlinear permutations (see Table 9.4) generated by

$$GF(2^4) = \frac{\mathbb{Z}_2[X]}{< x^4+x+1 >} = \left\{d_0 + d_1x + d_2x^2 + d_3x^3 : \ d_i \in \mathbb{Z}_2\right\}. \tag{9.35}$$

The multiplicative inverse is then transformed using the following affine transformation:

$$S - box_{_{AES}} = G \circ L \circ I, \tag{9.36}$$

where $L(x)$ is the linear mapping, $I(x)$ is inverse functions that gives inverse of nonzero elements of Galois field and zero is mapped to itself and G is translational function that is $G(x) = x \oplus b$. The circulant matrix over $GF(2)$ is of the form

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}. \tag{9.37}$$

Finally, we have obtained four different S-boxes which are given as follows:

Table 9.5: Proposed S-boxes based on nonlinear permutations.

$j \ddots i$	**0**	**1**	**2**	**3**
0	11	0	10	7
1	2	9	8	4
2	13	15	12	3
3	5	1	6	14

S-box-1

$j \ddots i$	**0**	**1**	**2**	**3**
0	11	0	15	10
1	12	7	3	2
2	5	9	1	8
3	6	4	14	13

S-box-2

$j \ddots i$	**0**	**1**	**2**	**3**
0	11	0	9	15
1	1	10	8	12
2	6	7	4	3
3	14	2	13	5

S-box-3

$j \ddots i$	**0**	**1**	**2**	**3**
0	11	0	7	9
1	4	15	3	1
2	14	10	2	8
3	13	12	5	6

S-box-4

9.4.2 Least Significant Bits

In computing, the least significant bit (lsb or LSB) is the bit position in a balancing whole number giving the units esteem, that is, figuring out if the number is even or odd. The lsb is at times alluded to as the right-most bit, because of the meeting in positional documentation of composing less significant digit further to the right. It is similar to the least significant digit of a decimal number, which is the digit in the ones (right-most) position [165].

9.4.3 Most Significant Bits

The most significant bit (msb or MSB, likewise called the high-order bit) is the bit position in a parallel number having the highest value. The msb is in some cases suggested to as the left-most bit because of the assembly in positional documentation of composing more significant digits further to the left [165].

9.4.4 Chaotic Gauss Map

The Gauss map (also known as Gaussian map or mouse map), is a nonlinear iterated map of the real into a real interval given by the Gaussian function:

$$x_{n+1} = e^{-\gamma x_n^2} + \delta, \tag{9.38}$$

where γ and δ are real parameters. Named after Johann Carl Friedrich Gauss, the function maps the bell shaped Gaussian function similar to the logistic map. In the parameter real space x_n can be chaotic.

For $\gamma = 0.62$ and $\delta \in [-1, 1]$ with initial approximation $x_0 = 0.1$, this maps exhibits chaotic behavior [176].

9.4.5 Chaotic Logistic Map

A simple system that is useful to illustrate some properties of chaotic systems is the logistic map. This is a non-linear recurrence relation with a single control parameter r,

$$x_{n+1} = r(1 - x_n). \tag{9.39}$$

Near $r = 3.59$, the system becomes chaotic. There is no periodicity in the limiting behavior. Furthermore for those values of r for which the system is chaotic, the sequence of x_n generated by the logistic map depends sensitively on the beginning value x_0 [176].

9.4.6 Proposed Watermarking Scheme

In this section, we have implemented the proposed algebraic structure to watermarking. Our main purpose here is to hide an invisible watermarked with the help of proposed S-boxes and least significant digits. The algorithm and flow chart of the proposed watermarking scheme is given as follows:

Algorithm

1. *Take an image in which watermarked is to be embedded.*
2. *Transform values of each pixel into an array of eight bits.*
3. *Separate MSBs and LSBs for each pixel of test image.*
4. *Using chaotic Gauss map to locate position of LSB where mark is to be placed.*
5. *Apply chaotic Logistic map that decide which proposed S-box is used for watermark.*
6. *Apply S-box transformation on LSBs that signify the position of values in selected S-box that has to be replacing with binaries of LSBs.*
7. *Repeat steps 4-6, until the whole image is replaced.*
8. *Lastly, rebuild MSBs and transform LSBs.*

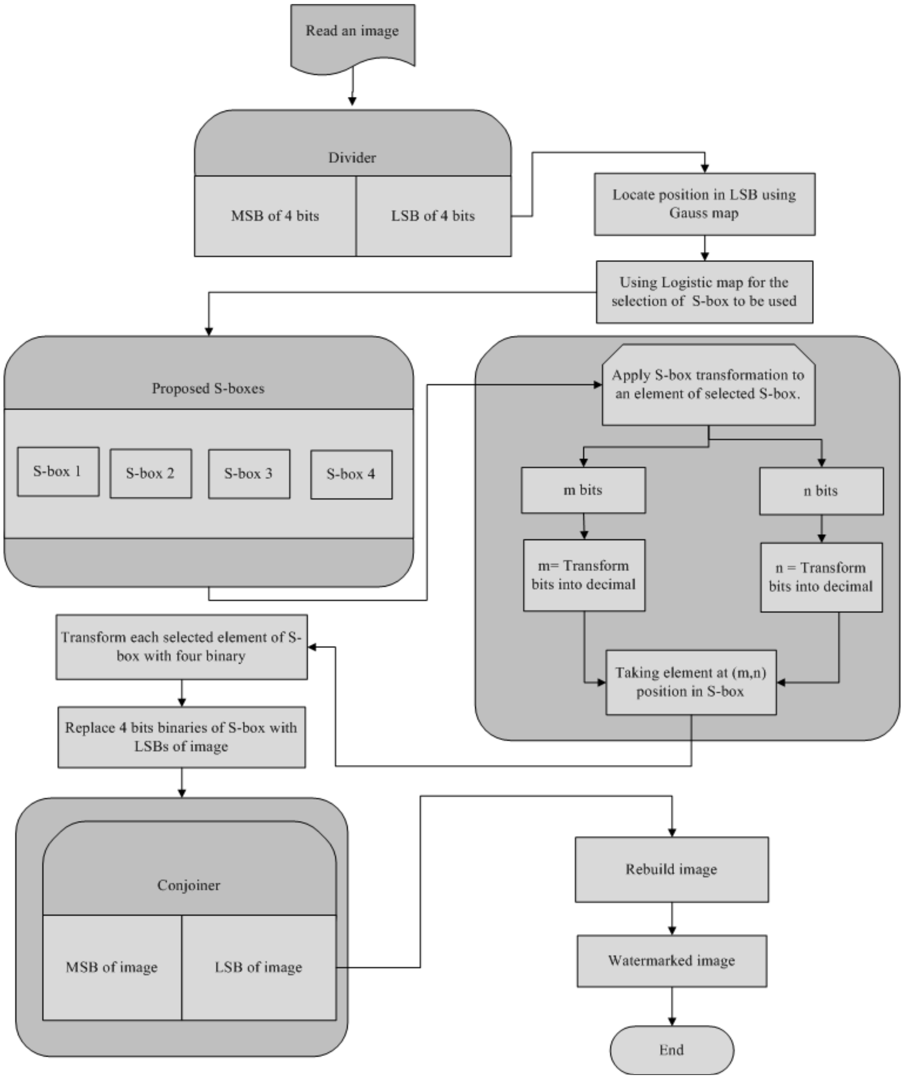
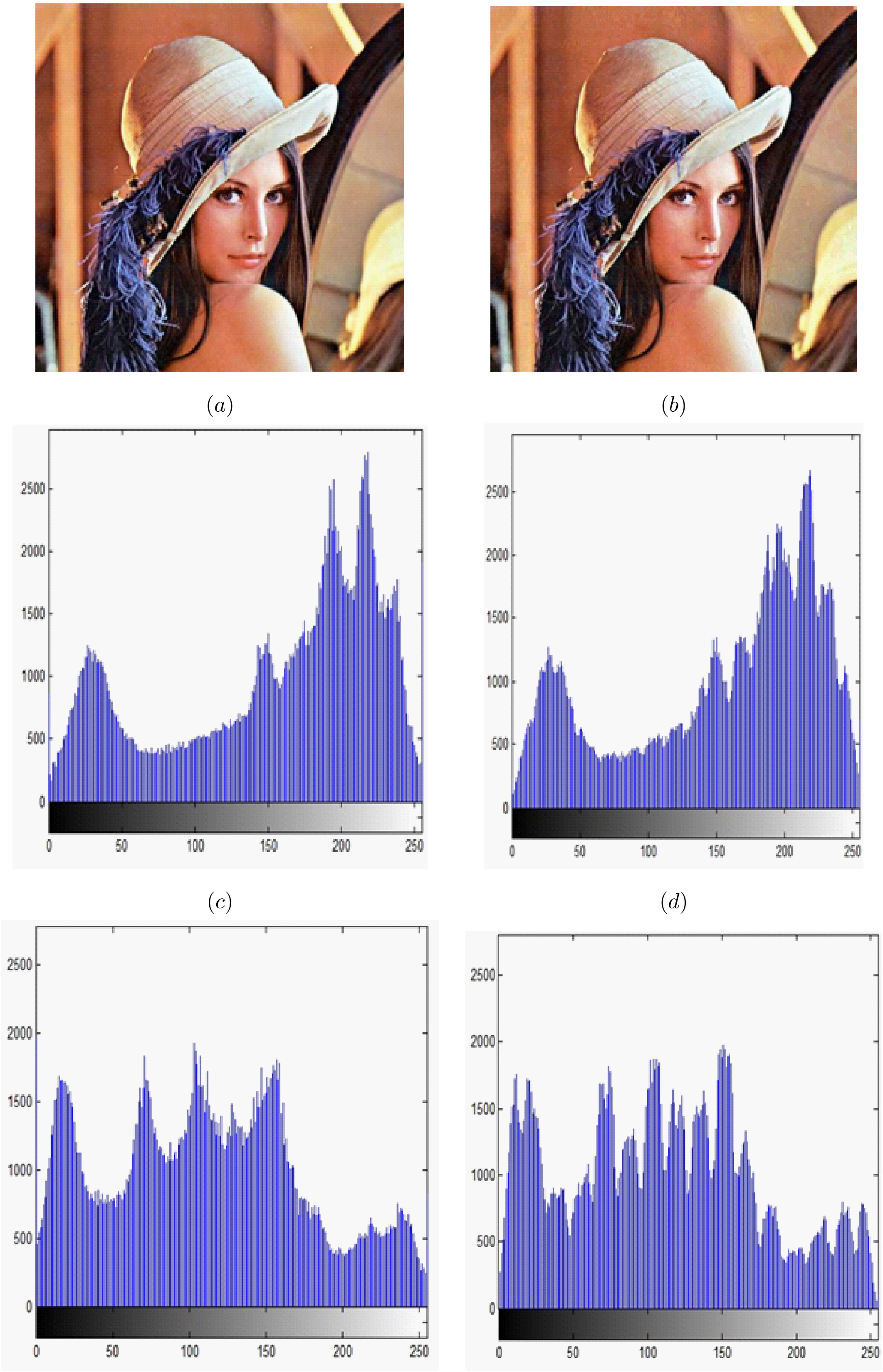
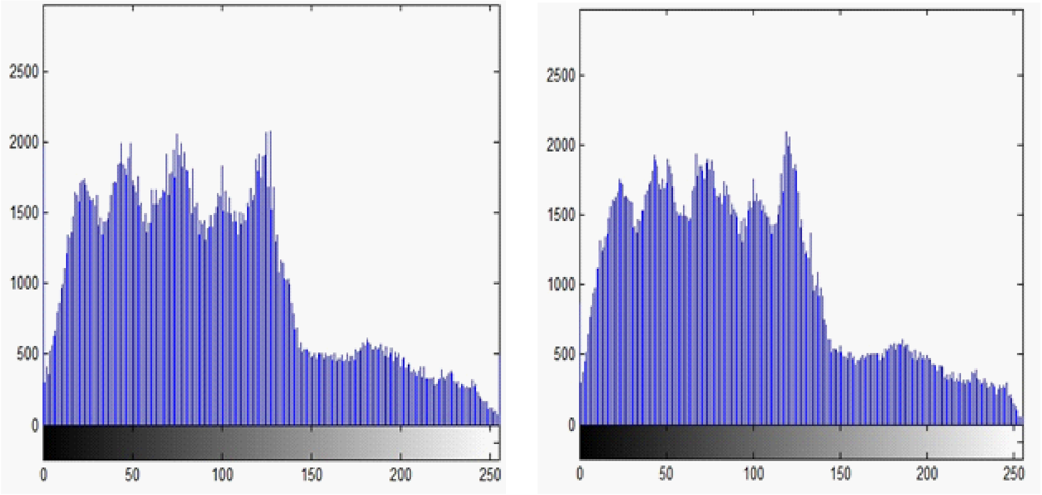

Read an image
Divider
MSB of 4 bits
LSB of 4 bits
Locate position in LSB using Gauss map
Using Logistic map for the selection of S-box to be used
Proposed S-boxes
S-box 1
S-box 2
S-box 3
S-box 4
Apply S-box transformation to an element of selected S-box.
m bits
n bits
m= Transform bits into decimal
n = Transform bits into decimal
Taking element at (m,n) position in S-box
Transform each selected element of S-box with four binary
Replace 4 bits binaries of S-box with LSBs of image
Conjoiner
MSB of image
LSB of image
Rebuild image
Watermarked image
End

Fig. 9.1: Flow chart of proposed copyright protection algorithm.


(a)
(b)
2500
2000
1500
1000
500
0
0
50
100
150
200
250
(c)
(d)
(e)
(f)



(g) (h)

Fig. 9.2: (a) Original Lena image of size,(b)Watermarked image of size,(b-e-g) Red, Green and Blue components of original image,(c-f-g) Red, Green and Blue components of watermarked image.

9.4.7 Performance Evaluation Metric

In this section, we have utilized proposed performance analysis for watermarked images. This section mainly described computational results which authenticate the insertion of watermarked inside the original image. The first distance metric which is Euclidean distance clearly shows that there is a watermarked inside the image after applying the proposed algorithm. As the distance value is 11.68380000 which is only possible when some of the pixels intensities differ from original one. In a similar fashion, SED, NSED, MD, CD and DC clearly reveal the difference between original and watermarked images. In term of probability distribution, MIV and NMIV values for original and watermarked images authenticate the embedding of watermark. With the help of these proposed quality metrics, we can easily draw significant difference between host image (original data) and watermarked image without plotting histogram analysis.

Table 9.6: Metric distances for original and watermarked images with proposed scheme.

Proposed features	Numerical results for proposed features
ED	11.68380000
SED	136.5120000
NSED	0.000576846
MD	7643.130000
CD	0.000293304
DC	0.001149590
MIV(original image)	0.868448000
MIV(watermarked image)	0.449630000
NMIV(original image)	0.614551000
NMIV(watermarked image)	0.376516000

9.5 Conclusion

There are different techniques used in watermarking for security of images which based on frequency domain, spatial domain and spread spectrum algorithm. In this chapter, we have used spatial domain method that based on LSB for security of images, which is easy, simple and more effective method. The proposed scheme is robust due to its algebraic properties where, we have used small Galois field to construct S-boxes which are hold onto any secure communication. The main purpose here to reduce the computational complexity that is involved in large size S-boxes. Moreover, we have presented a novel statistical procedure for testing watermarked images which is yet not been presented in literature.

Chapter 10

Construction of Substitution Boxes over the Classes of Chain Rings

The meanings of passing information from one side to other side by a conventional way is been changed because of Internet and Communication Technology. The issues of the security and the uprightness of information increase due to fast developments in digital world. Presently digital communication has become an important part of transmission of information securely. There are various Internet applications which are utilized to convey covertly. As an outcome, the security of data against unapproved access has turned into a prime target. This leads to parts of advancement of different systems for information hiding. Cryptography and Watermarking are famous techniques for hiding information accessible to conceal information safely. Our main goal here is to develop innovative algorithms for information hiding which includes cryptography, watermarking and steganography but we will concentrate on first two in this chapter. Moreover, we construct novel S-boxes which is based on finite chain rings and apply statistical analyses to examine the strength of proposed algorithms of image encryption and watermarking.

10.1 Galois Rings and their Groups of Unit Elements

In this section, we discuss some elementary concepts, for instance; Local commutative ring with identity, Galois extension ring, unit elements, and maximal cyclic subgroup of group of invertible elements of a Galois ring.

10.1.1 Galois Rings

We begin with some basic definitions of unitary (local) commutative rings.

Definition 102 *Let R be a commutative ring with unity. An element u is unit in R if there exists an element v in R such that $u.v = 1$, where 1 is the identity of R.*

Definition 103 *A commutative ring R with unity is said to be local if and only if its all non-unit elements form an additive Abelian group. For instance $\mathbb{Z}_{p^k}$, p is a prime integer and k is any positive integer, is a local ring.*

Definition 104 *Let R be a commutative ring with unity. A non-zero element a is a zero divisor in R if there exists a non-zero element b in R such that $a.b = 0$.*

Definition 105 *Let (R, M) be a local commutative ring with unity. An irreducible polynomial $f(x) \in R[x]$ over R is said to be a basic irreducible polynomial if it is irreducible over the corresponding residue field K, where $(K = R/M)$.*

Consider the finite local ring $\mathbb{Z}_{p^k}$, where p is prime and k is a positive integer with corresponding residue field $\mathbb{Z}_p$. Now $\mathbb{Z}_{p^k}[x] = \{a_0 + a_1x + a_2x^2 + ... + a_nx^n : a_i \in \mathbb{Z}_{p^k}, n \in \mathbb{Z}^+\}$ is the polynomial extension of $\mathbb{Z}_{p^k}$ in the variable x and $\mathbb{Z}_p[x] = \{a_0 + a_1x + a_2x^2 + ... + a_nx^n : a_i \in \mathbb{Z}_p, n \in \mathbb{Z}^+\}$ is the polynomial extension of $\mathbb{Z}_p$ in the variable x. Let $f(x) \in \mathbb{Z}_{p^k}[x]$ be a basic irreducible polynomial with degree h. Ideal generated by $f(x)$ is denoted as $\langle f(x)\rangle$ and defined as $\langle f(x)\rangle = \{a(x).f(x) : a(x) \in \mathbb{Z}_{p^k}[x]\}$. Let $R = \frac{\mathbb{Z}_{p^k}[x]}{\langle f(x)\rangle} = \{a_0 + a_1x + a_2x^2 + ... + a_{h-1}x^{h-1} : a_i \in \mathbb{Z}_{p^k}\}$ denote the set of residue classes of polynomial in x over $\mathbb{Z}_{p^k}$, modulo the polynomial $f(x)$. This ring, denoted by $GR(p^k, h)$, is a commutative ring with identity and is called the Galois extension of $\mathbb{Z}_{p^k}$. Also $GR(p^k, 1)$ is isomorphic to $\mathbb{Z}_{p^k}$, and $GR(p, h) = \frac{\mathbb{Z}_{p^k}[x]}{\langle \overline{f}(x)\rangle} = K$ is isomorphic to $GF(p^h)$, a Galois field extension of $\mathbb{Z}_p$ having p^h elements, where $\overline{f} = r_p(f)$ polynomial f which has coefficient modulo p [196]-[198].

10.1.2 Maximal Cyclic Subgroup of Group of Units of Galois Rings

Let K^* and R^* be the multiplicative group of units of field and the ring K and R, respectively. Then R^* is an Abelian group and can be written in the direct product of cyclic subgroups. By the following Theorem from [1, Theorem 2], between these cyclic subgroups, there is only one cyclic subgroup of order $p^h - 1$.

Theorem 106 *R^* has one and only one cyclic subgroup of order relatively prime to p. This cyclic subgroup has order $p^h - 1$.*

The cyclic subgroup of order $p^h - 1$ can be generated by the generator of the corresponding finite field. This cyclic subgroup is denoted by G_n, where $n = p^h - 1$. Since the order of K^* and G_n is the same, i.e., $p^h - 1$ and both will be cyclic. Therefore G_n is isomorphic to K^*.

10.2 Construction of S-boxes based on Maximal Cyclic Subgroups

In order to create confusion in a data many techniques can be used to do so. One of these technique is using an S-box. The strongest S-boxes are constructed through mathematical formulas and systematic calculations. In order to improve the quality many have worked in the Galois fields $GF(2^n)$, $1 \leq n \leq 8$ and created numerous S-boxes. In [199], a 4×4 S-box over maximal cyclic subgroup of group of units of Galois ring $GR(4,4)$ is constructed with its application in watermarking. However, as an extension to [199], in this section a novel construction technique of 4×4 S-boxes with the utility of maximal cyclic subgroups of groups of units of the Galois rings $GR(4,4)$, $GR(8,4)$ and $GR(32,4)$ is given. While, in each three cases the maximal cyclic subgroups G_{15} of orders 15 are, respectively, isomorphic to the cyclic Galois group $GF(2,4)^*$. The association of maximal cyclic subgroups with admiring cyclic Galois group $GF(2,4)^*$, which are caused by the mod-2 reduction maps from local commutative rings $\mathbb{Z}_4$, $\mathbb{Z}_8$ and $\mathbb{Z}_{32}$ to their common residue field $\mathbb{Z}_2$, supports in construction of the 4×4 S-boxes over maximal cyclic subgroups. Of course these newly designed S-boxes are increasing complexity during encryption and decryption.

10.2.1 S-box Construction Algorithm on Galois Ring $GR(2^m, 4)$

Given below is the procedure, defining the S-box in 4 steps:

Step.1: Inversion function $I : G_n \cup \{0\} \rightarrow G_n \cup \{0\}$.

Step.2: Linear scalar multiple function $f : G_n \cup \{0\} \rightarrow G_n \cup \{0\}$.

Step.3: Take composition of $I \circ f$ to get $(n+1) \times (n+1)$ S-box.

Step.4: Apply permutations S_n to each element of S-box obtained in step 3, which gives us $n!$ S-boxes.

The map described above is nothing more than a substitution within the set $G_n \cup \{0\}$. An element of the set is substituted with the element next to its respective inverse. (In this case we define this direction with increasing power of the generator) or in other words the scalar multiplied with the inverse. In the examples below we discuss and analyze this construction method for S-boxes of size 4×4.

Let us consider the local rings $\mathbb{Z}_4 = \{0,1,2,3\}, \mathbb{Z}_8 = \{0,1,2,\ldots,7\}$ and $\mathbb{Z}_{16} = \{0,1,2,\ldots,15\}$, and $\mathbb{Z}_{32} = \{0,1,2,\ldots,31\}$, whereas $\mathbb{Z}_2 = \{0,1\}$, is their common residue field. The monic polynomial $f(x) = x^4 + x + 1$ is basic irreducible over the local rings $\mathbb{Z}_4$, $\mathbb{Z}_8$, $\mathbb{Z}_{16}$ and $\mathbb{Z}_{32}$ such that $f(x) = f(x) \ mod \ 2 = x^4 + x + 1$ is irreducible polynomial over $\mathbb{Z}_2$.

10.2.2 S-box on $GF(2^4)$

Take the polynomial ring $\mathbb{Z}_2[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in \mathbb{Z}_2, n \in Z^+\}$ in one indeterminate x over binary field $\mathbb{Z}_2$. Let $< f(x) >= \{a(x).f(x) : a(x) \in \mathbb{Z}_2[x]\}$ be the principal ideal in $\mathbb{Z}_2[x]$, generated by $f(x)$. Then elements of Galois extension field $K = \mathbb{Z}_2[x])/(< f(x) >)$, of order 16 are given in Table. 10.1.

Table 10.1: Elements of Galois field $GF(2^4)$.

Exp.	Polynomial	Binaries Representation	Exp.	Polynomial	Binaries Representation
$-\infty$	0	0000	7	$1 + x^2 + x^3$	1011
0	1	1000	8	x^2	0010
1	$1 + x$	1100	9	$x^2 + x^3$	0011
2	$1 + x^2$	1010	10	$1 + x + x^2$	1110
3	$1 + x + x^2 + x^3$	1111	11	$1 + x^3$	1001
4	x	0100	12	x^3	0001
5	$x + x^2$	0110	13	$1 + x + x^3$	1101
6	$x + x^3$	0101	14	$x + x^2 + x^3$	0111

Now, let us construct the S-box on the Galois field extension $GF(2^4)$ (see Table. 10.1). It can be seen in Table 10.2 that it is the most basic S-box and it satisfies all the fundamental properties being an S-box.

Table 10.2: S-box on $GF(2^4)$.

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

10.2.3 S-box on $GR(4, 4)$

Let $\mathbb{Z}_4[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in \mathbb{Z}_4, n \in Z^+\}$ is the polynomial ring with one indeterminate x and $< f(x) >= \{a(x).f(x) : a(x) \in \mathbb{Z}_4[x]\}$ is a principal ideal generated by $f(x)$. Thus $R = (\mathbb{Z}_4[x])/(< f(x) >) = \{a_0 + a_1x + a_2x^2 + \cdots a_{(4-1)}x^{(4-1)} : a_i \in \mathbb{Z}_4\}$ is the Galois ring extension of order 256 with corresponding Galois field extension $K = (\mathbb{Z}_2[x])/(< f(x) >)$ of order 16, whose elements are given in Table 1. $K^* = K\backslash\{0\}$ becomes the multiplicative group of units of the field K. Now, let R^* be the multiplicative group of units of the Galois ring R. Then the maximal cyclic subgroup of R^*, isomorphic to the cyclic Galois group K^*, of order 15 is denoted by G_{15} and it is given in Table 10.3.

Table 10. 3: Elements of $G_{15} \cup \{0\}$ in $GR(4,4)$.

Exp.	Polynomial		Exp.	Polynomial	
$-\infty$	0	0000	14	$x + 3x^2 + x^3$	0131
0	1	1000	16	$3 + 3x$	3300
2	$1 + 2x + x^2$	1210	18	$3 + x + x^2 + 3x^3$	3113
4	$3x + 2x^2$	0320	20	$x + 3x^2 + 2x^3$	0132
6	$2 + x + 3x^3$	2103	22	$1 + 3x^2 + x^3$	1031
8	x^2	0010	24	$3x^2 + 3x^3$	0033
10	$3 + 3x + x^2 + 2x^3$	3312	26	$3 + x^3$	3001
12	$2 + 2x + 3x^3$	2203	28	$1 + 3x + 2x^2 + x^3$	1321

Followed by the construction algorithm and using maximal cyclic subgroup of Table 10.3. We obtain S-box given in the Table 10.4.

Table 10.4: S-Box on $GR(4,4)$.

0	67	215	159
25	240	15	16
1	113	116	198
109	45	202	44

10.2.4 S-box on $GR(8,4)$

$\mathbb{Z}_8[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in \mathbb{Z}_8, n \in Z^+\}$ is the polynomial ring with one indeterminate x and $< f(x) >= \{a(x).f(x) : a(x) \in \mathbb{Z}_8[x]\}$ is principal ideal generated by $f(x)$. Thus $R = (\mathbb{Z}_8[x])/(< f(x) >) = \{a_0 + a_1x + a_2x^2 + \cdots a_{(4-1)}x^{(4-1)} : a_i \in \mathbb{Z}_8\}$ is the Galois ring extension of order 4096 with corresponding Galois field extension $K = (\mathbb{Z}_2[x])/(< f(x) >)$ of order 16, whose elements are given in Table 10.4. $K^* = K\backslash\{0\}$ becomes the multiplicative group the field K. Now, let R^* be the multiplicative group of units of R. Then the cyclic subgroup of R^*, isomorphic to K^*, of order 15 is denoted by G_{15}

and is given in Table 10.5.

Table 10.5: Elements of $G_{15} \cup \{0\}$ in $GR(8,4)$.

Exp.	Polynomial		Exp.	Polynomial	
$-\infty$	0	0000	14	$x + 7x^2 + x^3$	0171
0	1	1000	16	$7 + 7x$	7700
2	$1 + 2x + x^2$	1210	18	$7 + 5x + 5x^2 + 7x^3$	7557
4	$3x + 6x^2 + 4x^3$	0364	20	$4 + x + 7x^2 + 6x^3$	4176
6	$2 + x + 3x^3$	2103	22	$1 + 7x^2 + 5x^3$	1075
8	$4 + 4x + x^2 + 4x^3$	4414	24	$4x + 3x^2 + 3x^3$	0433
10	$3 + 7x + x^2 + 2x^3$	3712	26	$7 + 5x^3$	7005
12	$6 + 6x + 3x^3$	6603	28	$5 + 7x + 2x^2 + 5x^3$	5725

By using proposed S-box construction algorithm and using maximal cyclic subgroup of Table 10.5, we obtain S-box given in the Table 10.6.

Table 10.6: S-Box on $GR(8,4)$.

0	3	111	123
81	224	63	100
1	193	200	10
189	195	60	152

10.2.5 Nonexistence of S-box on $GR(16,4)$

$\mathbb{Z}_{16}[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in \mathbb{Z}_{16}, n \in Z^+\}$ is the polynomial ring with one indeterminate x and $< f(x) >= \{a(x).f(x) : a(x) \in \mathbb{Z}_{16}[x]\}$ is principal ideal generated by $f(x)$. Thus $R = (\mathbb{Z}_{16}[x])/(< f(x) >) = \{a_0 + a_1x + a_2x^2 + \cdots a_{(4-1)}x^{(4-1)} : a_i \in \mathbb{Z}_{16}\}$ is the Galois ring extension of order 65535 with corresponding Galois field extension $K = (\mathbb{Z}_2[x])/(< f(x) >)$ of order 16, whose elements are given in Table 10.7. $K^* = K \backslash \{0\}$ becomes the multiplicative group the field K. Now, let R^* be the multiplicative group of units of R. Then the cyclic subgroup of R^*, isomorphic to K^*, of order 15 is denoted by G_{15}

and is given in Table 10.7.

Table 10.7: Elements of $G_{15} \cup \{0\}$ in $GR(16,4)$.

Exp.	Polynomial		Exp.	Polynomial	
$-\infty$	0	0000	14	$9x + 15x^2 + x^3$	$09F1$
0	1	1000	16	$15 + 7x + 8x^3$	$F708$
2	$1 + 2x + x^2$	1210	18	$15 + 13x + 5x^2 + 15x^3$	$FD5D$
4	$3x + 6x^2 + 4x^3$	0364	20	$12 + 9x + 15x^2 + 6x^3$	$C9F6$
6	$2 + x + 8x^2 + 3x^3$	2183	22	$1 + 7x^2 + 13x^3$	$107D$
8	$4 + 4x + 9x^2 + 4x^3$	4494	24	$4x + 11x^2 + 11x^3$	$04BB$
10	$3 + 7x + x^2 + 10x^3$	$371A$	26	$15 + 8x + 8x^2 + 5x^3$	$F885$
12	$14 + 14x + 8x^2 + 3x^3$	$EE83$	28	$13 + 15x + 2x^2 + 13x^3$	$DF2D$

Followed by the construction algorithm and using maximal cyclic subgroup of Table 10.7, we obtain S-box given in the Table 10.8.

Table 10.8: S-Box on $GR(16,4)$.

0	143	223	115
33	64	127	68
1	1	144	18
253	156	238	48

The structure Table 10.8 is not an S-Box as repetition of 1 on two positions. So, this gives us a counter example that, not every maximal cyclic subgroup of the group of units of Galois ring extension generates an S-box.

10.2.6 S-box on $GR(32,4)$

$\mathbb{Z}_{32}[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : a_i \in Z_{32}, n \in Z^+\}$ is the polynomial ring with one indeterminate x and $< f(x) >= \{a(x).f(x) : a(x) \in \mathbb{Z}_{32}[x]\}$ is principal ideal generated by $f(x)$. Thus $R = (\mathbb{Z}_{32}[x])/(< f(x) >) = \{a_0 + a_1x + a_2x^2 + \cdots a_{(h-1)}x^{(h-1)} : a_i \in \mathbb{Z}_{32}\}$ is the Galois ring extension of order 1048576 with corresponding Galois field extension $K = (\mathbb{Z}_2[x])/(< f(x) >)$ of order 16, whose elements are given in Table 10.9. $K^* = K\backslash\{0\}$ becomes the multiplicative group the field K. Now, let R^* be the multiplicative group of units of R. Then the cyclic subgroup of R^*, isomorphic to K^*, of order 15 is denoted by G_{15} and is given in Table 10.9.

Table 10.9: Elements of $G_{15} \cup \{0\}$ in $GR(32, 4)$.

Exp.	Polynomial		Exp.	Polynomial	
$-\infty$	0	0000	28	$13 + 15x + 18x^2 + 13x^3$	$DFID$
0	1	1000	32	$17 + 2x + 17x^2 + 16x^3$	$H2HG$
4	$3x + 6x^2 + 4x^3$	0364	36	$2 + 17x + 8x^2 + 3x^3$	$2H83$
8	$4 + 20x + 9x^2 + 20x^3$	$4K9K$	40	$3 + 23x + x^2 + 26x^3$	$2N1Q$
12	$30 + 14x + 8x^2 + 19x^3$	$UE8J$	44	$16 + 25x + 15x^2 + 17x^3$	$GPFH$
16	$31 + 7x + 24x^3$	$V700$	48	$15 + 29x + 5x^2 + 31x^3$	$FT5V$
20	$28 + 9x + 31x^2 + 6x^3$	$S9V6$	52	$17 + 16x + 7x^2 + 29x^3$	$HG7T$
24	$16 + 4x + 11x^2 + 11x^3$	$G4BB$	56	$31 + 8x + 24x^2 + 5x^3$	$V8O5$

Followed by the construction algorithm and using maximal cyclic subgroup of Table 10.9, we obtain S-box given in the Table 10.10.

Table 10.10: S-Box on $GR(32, 4)$.

0	17	34	60
96	175	81	255
1	48	237	222
31	227	144	132

So, we are not certain if G_s of every Galois ring will generate an S-box for us. This implies that with a certain polynomial and Galois ring structure we are not sure if we will get an S-box over it or not. It shows that, the method discussed in [11] is not an efficient technique to get S-boxes for use in different applications. Even though these newly designed S-boxes are increasing encryption and decryption difficulty as compare to the S-boxes constructed over Galois field $GF(2, 4)$.

10.3 Basic Preliminaries of Finite Chain Ring of the Type $\frac{\mathbb{F}_2[u]}{<u^k>} = \mathbb{F}_2 + u\mathbb{F}_2 + \cdots + u^{k-1}\mathbb{F}_2$

Let $\mathcal{R}$ be a ring. An element v is unit in $\mathcal{R}$ if there exists an element w in $\mathcal{R}$ such that $vw = 1$, where 1 is the identity of $\mathcal{R}$. Unit elements of a ring form a multiplicative group. A non-zero element a is a zero divisor in $\mathcal{R}$ if there exists a non-zero element b in $\mathcal{R}$ such that $ab = 0$. A nonzero element a is said to be nilpotent element in $\mathcal{R}$ if there exists a positive integer k such that $a^k = 0$. The least positive integer k with this property is known as the nilpotency index a.

A ring $\mathcal{R}$ is local if and only if its all non-unit elements form an additive Abelian group. More unambiguously a local ring $\mathcal{R}$ has a unique maximal ideal $\mathcal{M}$ and the factor ring $\frac{\mathcal{R}}{\mathcal{M}}$ is its residue field.

A local finite ring $\mathcal{R}$ is a chain ring if and only if the radical $\mathcal{M}$ of $\mathcal{R}$ is a principal ideal (consists of all multiples of a fixed element of $\mathcal{R}$, and this fixed element is called the generator of the ideal), and therefore the factor ring $\frac{\mathcal{R}}{\mathcal{M}}$ is a field. Thus ideals of a chain ring form a chain. The famous examples of such rings are; $\mathbb{Z}_{p^n}$, the ring of integers modulo p^n where p is prime, and the Galois field $GF(p^n) = \mathbb{F}_q$ with $q = p^n$ elements. Another large class of finite chain rings is the Galois rings $GR(p^n, r) = \frac{\mathbb{Z}_{p^n}[x]}{<f(x)>}$, where $f(x) \in \mathbb{Z}_{p^n}[x]$ is monic irreducible polynomial of degree r generates the principal ideal $< f(x) >$, however $f(x)$ is also irreducible modulo the prime p, i.e. $f(x)$ is the basic irreducible polynomial. Whereas the Galois ring $\mathcal{R}$ =$GR(p^n, r)$ has p^{nr} number of elements and an element $\overline{a}(x)$ in $GR(p^n, r)$ has the representation $\overline{a}_0 + \overline{a}_1 x + ... + \overline{a}_{r-1}x^{r-1}, \overline{a}_0, \overline{a}_1, ..., \overline{a}_{r-1} \in \mathbb{Z}_{p^n}$. The radical $\mathcal{M}$ is the set of nilpotent elements of $\mathcal{R}$ and the residue field $\frac{\mathcal{R}}{\mathcal{M}}$ of $\mathcal{R}$ is the Galois extension field $GF(p^r)$. One of the typical class of chain rings is the factor ring $\frac{GF(p^r)[x]}{<x^k>}$ of Euclidean domain $GF(p^r)[x]$. The finite chain ring $\frac{GF(p^r)[x]}{<x^k>}(= \frac{\mathbb{F}_{p^r}[x]}{<x^k>})$ has the representation $\mathbb{F}_{p^r} + x\mathbb{F}_{p^r} + \cdots + x^{k-1}\mathbb{F}_{p^r}$.

Let R_k be the representation of finite chain ring $\frac{\mathbb{F}_2[u]}{<u^k>} = \mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2 + \cdots + u^{k-1}\mathbb{F}_2$. The ring R_k has 2^k number of elements. The element u is the nilpotent element with nilpotency index k (i.e., $u^k = 0$). Thus it follows that $< 0 >= u^k R_k \subset u^{k-1}R_k \subset \cdots \subset u^2 R_k \subset uR_k \subset R_k$ is the ascending chain of ideals in R_k and therefore R_k is a local ring with only maximal ideal uR_k, whereas, $\frac{R_k}{uR_k} \simeq \mathbb{F}_2$ is the residue field of the chain ring R_k. The ideals $u^i R_k$ and $u^{i+1}R_k$, where $i = 0, 1, 2, \cdots, k-1$, respectively have the cardinality 2^{k-i} and 2^{k-i+1}. Thus the cardinality of $u^i R_k$ is 2 times the cardinality of $u^{i+1}R_k$.

Amongst the rings of four elements, earlier the Galois field $\mathbb{F}_4$, and later the integers modulo 4 ring $\mathbb{Z}_4$, are frequently used in algebraic coding theory. Recently, Abualrub and Siap [200] studied cyclic codes of an arbitrary length n over the rings $\mathbb{F}_2 + u\mathbb{F}_2 = \{0, 1, u, \overline{u} = 1 + u\}$, with $u^2 = 0$, and $\mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2 = \{0, 1, u, u^2, 1 + u, 1 + u^2, u + u^2, 1 + u + u^2\}$, with $u^3 = 0$. However, Al-Ashker and Hamoudeh [203] extend these results to more general rings of the form $R_k = \mathbb{F}_2 + u\mathbb{F}_2 + \cdots + u_2^{k-1}\mathbb{F}_2$, with $u^k = 0$. The ring $\mathbb{F}_2 + u\mathbb{F}_2$ share some good properties of both $\mathbb{Z}_4$ and $\mathbb{F}_4$. The alphabet in the ring $\mathbb{F}_2 + u\mathbb{F}_2$ is given to all binary polynomials in indeterminate u of degree at most 1, and is closed under binary polynomial addition and multiplication modulo u^2. The multiplication and addition tables for the ring $\mathbb{F}_2 + u\mathbb{F}_2$ are given in Tables 10.11. The multiplication table of the ring $\mathbb{F}_2 + u\mathbb{F}_2$coincides with that of $\mathbb{Z}_4$, when u and $\overline{u}$ are replaced by 2 and 3 respectively. In this sense $\mathbb{F}_2 + u\mathbb{F}_2$ is analogous to $\mathbb{Z}_4$ and here u plays the role of 2. Whereas the addition table is different and is similar to that of the Galois field $\mathbb{F}_4 = \{0, 1, \beta, \beta^2 = 1 + \beta\}$, where $\overline{u}$ and u are replaced by β and β^2, respectively.

Table 10.11: × and + Tables for $\mathbb{F}_2 + u\mathbb{F}_2$.

$\times$	0	1	u	$\overline{u}$
0	0	0	0	0
1	0	1	u	$\overline{u}$
u	0	u	0	u
$\overline{u}$	0	$\overline{u}$	u	1

$+$	0	1	$\overline{u}$	u
0	0	1	$\overline{u}$	u
1	1	0	u	$\overline{u}$
$\overline{u}$	$\overline{u}$	u	0	1
u	u	$\overline{u}$	1	0

10.4 Construction of S-box Using Finite Chain Rings $\mathbb{F}_2 + u\mathbb{F}_2 + \cdots + u^{k-1}\mathbb{F}_2$

The chain ring $R_k = \frac{\mathbb{F}_2[u]}{<u^k>} = \mathbb{F}_2 + u\mathbb{F}_2 + ... + u^{k-1}\mathbb{F}_2$ has cardinality 2^k. As u is a nilpotent element with nilpotency index k, it follows that $< 0 >= u^k R_k \subset u^{k-1}R_k \subset ... \subset uR_k \subset R_k$. Accordingly the residue field of R_k is $\frac{R_k}{uR_k} \simeq \mathbb{F}_2$. The ring R_k shares some properties of the local ring $\mathbb{Z}_{2^k}$ and the Galois field $\mathbb{F}_{2^k}$. More explicitly the multiplication binary operation of R_k coincides with of $\mathbb{Z}_{2^k}$, whereas the addition binary operation is similar to that of $\mathbb{F}_{2^k}$. A significant S-box with wide-ranging cryptographic features is of ultimate worth for the development of resilient cryptographic system. Constructing cryptographically strong S-boxes is a basic challenge. In this study we propose a method to amalgam an efficient 4 × 4 S-box based on unit elements of the chain rings $\mathbb{F}_2 + u\mathbb{F}_2 + \cdots + u^{k-1}\mathbb{F}_2$. For the purpose we fix k to $2, 3, 4, 5, 6, 7$ and 8. The 4 × 4 S-box construction steps are given bellow:

Step.1: Take the multiplicative group of unit elements of the ring R_k which is given in table M_{G_k},

Step.2: If the cardinality of M_{G_k} is a perfect square and less than or equal to 16, define an inversion map $f : M_{G_k} \to M_{G_k}$ and a linear scalar multiple function $g : M_{G_k} \to M_{G_k}$. Otherwise choose a subgroup H_{G_k} of M_{G_k} of desired size 16 and then define these two bijective maps f and g from H_{G_k} to H_{G_k}. The selection of subgroups and defined maps for each ring are explicitly explained in subsections.

Step.3: Take the composition of the maps f and g.

Step.4: Generate 4 × 4 S-box by arranging them row wise.

Step.5: Apply permutations S_n to each elements of S-box obtained in step 4 which result in $n!$ S-boxes.

10.4.1 Construction of S-box through Multiplicative Group of R_3

The chain ring $R_3 = \frac{\mathbb{F}_2[u]}{<u^3>} = \mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2$ has 8 number of elements. The chain of ideals of this ring is $< 0 >= u^3R_3 \subset u^2R_3 \subset uR_3 \subset R_3$ and $\frac{R_3}{uR_3} \simeq \mathbb{F}_2$ is its residue field. The multiplication binary operation of R_3 coincides with of $\mathbb{Z}_8$, whereas the addition binary operation is similar to that of $\mathbb{F}_8$.

Table 10.12: Elements in chain ring R_3.

S. No.	Polynomial	Binary string	Sr. No.	Polynomial	Binary string
1	0	000	5	$1+u$	110
2	1	100	6	$1+u^2$	101
3	u	010	7	$u+u^2$	011
4	u^2	001	8	$1+u+u^2$	111

The multiplicative group of unit elements of the ring R_3 is

$$M_{G_3} = \{1, 1+u, 1+u^2, 1+u+u^2\}.$$

Define $f : M_{G_3} \to M_{G_3}$ by $f(a) = a^{-1}$ and $g : M_{G_3} \to M_{G_3}$ by $g(a) = a^{'}a$, where $a^{'} = 1+u$. Thus $f \circ g(a) = \left(a^{'}a\right)^{-1}$.

Table 10.13: Elements in $f \circ g(M_{G_3})$.

S. No.	Polynomial
$f \circ g(2)$	111
$f \circ g(5)$	101
$f \circ g(6)$	110
$f \circ g(8)$	100

Table 10.14: S-box over $R_3 = \mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2$.

7	5	6	4

10.4.2 Construction of S-box through Multiplicative Group of R_4

The chain ring $R_4 = \frac{\mathbb{F}_2[u]}{<u^4>} = \mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2 + u^3\mathbb{F}_2$ has 16 elements. Its chain of ideals is $< 0 >= u^4R_4 \subset u^3R_4 \subset u^2R_4 \subset uR_4 \subset R_4$,whereas the residue field of this ring is $\frac{R_4}{uR_4} \simeq \mathbb{F}_2$.The ring R_4 shares some properties of the local ring $\mathbb{Z}_{16}$ and the Galois field $\mathbb{F}_{16}$. The multiplication and addition binary operations of R_4 coincides with $\mathbb{Z}_{16}$ and $\mathbb{F}_{16}$ respectively.

Table 10.15: Elements in chain ring R_4.

S. No.	Polynomial	Binary string	S. No	Polynomial	Binary string
1	0	0000	9	$u+u^2$	0110
2	1	1000	10	$u+u^3$	0101
3	u	0100	11	u^2+u^3	0011
4	u^2	0010	12	$1+u+u^2$	1110
5	u^3	0001	13	$1+u+u^3$	1101
6	$1+u$	1100	14	$1+u^2+u^3$	1011
7	$1+u^2$	1010	15	$u+u^2+u^3$	0111
8	$1+u^3$	1001	16	$1+u+u^2+u^3$	1111

Multiplicative group of unit elements of the ring R_4 is

$$M_{G_4} = \{1, 1+u, 1+u^2, 1+u^3, 1+u+u^2, 1+u+u^3, 1+u^2+u^3, 1+u+u^2+u^3\}.$$

Take a subgroup $H_{G_4} = \{1, 1+u, 1+u^2, 1+u+u^2+u^3\}$ of index 2 of the group M_{G_4} and apply given procedure on subgroup rather than group M_{G_4}. Define $f : H_{G_4} \to H_{G_4}$ by $f(a) = a^{-1}$ and $g : H_{G_4} \to H_{G_4}$ by $g(a) = a^{'}a$, where $a^{'} = 1+u$, $f \circ g(a) = \left(a^{'}a\right)^{-1}$. The following Table 10.16 is of $f \circ g(H_{G_4})$ in binary and decimal form, which is in fact the S-box constructed over the chain ring $R_4 = \mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2 + u^3\mathbb{F}_2$.

Table 10.16: S-box over R_4.

15	10	12	8

10.4.3 Construction of S-box through Multiplicative Group of R_5

The chain ring $R_5 = \frac{\mathbb{F}_2[u]}{<u^5>} = \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2$ has 32 number of elements. The chain of ideals is, $< 0 >= u^5R_5 \subset u^4R_5 \subset u^3R_5 \subset u^2R_5 \subset uR_5 \subset R_5$ and its residue field is $\frac{R_5}{uR_5} \simeq \mathbb{F}_2$.The multiplication binary operation of R_5 coincides with of $\mathbb{Z}_{2^5}$, whereas the addition binary operation is similar to that of $\mathbb{F}_{2^5}$. Multiplicative group of unit elements of the ring R_5 is

$$\begin{aligned} M_{G_5} \;=\; & \{1, 1+u, 1+u^2, 1+u^3, 1+u^4, 1+u+u^2, 1+u+u^3, 1+u+u^4, 1+u^2+u^3, \\ & 1+u^2+u^4, 1+u^3+u^4, 1+u+u^2+u^3, 1+u+u^2+u^4, 1+u+u^3+u^4, \\ & 1+u^2+u^3+u^4, 1+u+u^2+u^3+u^4\}. \end{aligned}$$

Define $f : M_{G_5} \to M_{G_5}$ by $f(a) = a^{-1}$ and $g : M_{G_5} \to M_{G_5}$ by $g(a) = a'a$, where $a' = 1+u$. Thus $(f \circ g)(a) = (a'a)^{-1}$. The following Table 10.17 is of $f \circ g(H_{G_5})$ in binary and decimal form, which is in fact the S-box constructed over the chain ring $R_5 = \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2$.

Table 10.17: S-box over R_5.

31	30	26	19
21	18	24	23
25	22	29	27
28	20	17	16

10.4.4 Construction of S-box through Multiplicative Group of R_6

The chain ring $R_6 = \mathbb{F}_2[u]/ < u^6 >= \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2 + u^5\mathbb{F}_2$ has cardinality 64. As u is a nilpotent element with nilpotency index 6, it follows that $< 0 >= u^6R_6 \subset u^5R_6 \subset u^4R_6 \subset u^3R_6 \subset u^2R_6 \subset uR_6 \subset R_6$ and the residue field of R_6 is $\frac{R_6}{uR_6} \simeq \mathbb{F}_2$. The addition and multiplication binary operation of R_6 coincides with $\mathbb{F}_{2^6}$ and $\mathbb{Z}_{2^6}$ respectively. Multiplicative group of the ring R_6 is

$$\begin{aligned} M_{G_6} \;=\; & \{1, 1+u, 1+u^2, 1+u^3, 1+u^4, 1+u^5, 1+u+u^2, 1+u+u^3, 1+u+u^4, 1+u+u^5, \\ & 1+u^2+u^3, 1+u^2+u^4, 1+u^2+u^5, 1+u^3+u^4, 1+u^3+u^5, 1+u^4+u^5, 1+u+u^2+u^3, \\ & 1+u+u^2+u^4, 1+u+u^2+u^5, 1+u+u^3+u^4, 1+u+u^3+u^5, 1+u+u^4+u^5, \\ & 1+u^2+u^3+u^4, 1+u^2+u^3+u^5, 1+u+u^2+u^3+u^4, 1+u+u^2+u^3+u^5, 1+u+u^2+u^4+u^5, \\ & 1+u+u^3+u^4+u^5, 1+u^2+u^3+u^4+u^5, 1+u+u^2+u^3+u^4+u^5\}. \end{aligned}$$

The multiplicative subgroup M_{G_6} contains 32 elements, sixteen elements of order 8, 8 elements of order 4, 7 elements of order 2, and one element of order 1. Since our interest is in the subgroups of cardinality 16, so we combine these cyclic subgroups in such a way that they generate subgroups of order 16. The availability of subgroups of cardinality 16 is as follows:

Remark 107 *(i) Product of* 2 *elements of order* 4.

(ii) Product of 2 *elements of order* 2 *and* 1 *element of order* 4.

(iii) Product of 1 *element of order* 2 *and* 1 *element of order* 2.

In all of the above mentioned products, intersection of each joining pair or triplet should be just the identity element. We take one of these subgroups, $H_{G_6} = \langle 1+u^2, 1+u^3+u^4, 1+u^3+u^5 \rangle$ of cardinality 16 of the multiplicative group M_{G_6}. Define the maps $f : H_{G_6} \to H_{G_6}$ by $f(a) = a^{-1}$ and $g : H_{G_6} \to H_{G_6}$ by $g(a) = a'a$, where $a' = 1 + u^4$. Thus, $(g \circ f)(a) = (a^{'}a)^{-1}$. The following Table 11.18 is of $f \circ g(H_{G_6})$

in binary and decimal form, which is in fact the S-box designed over the chain ring R_6.

Table 10.18: S-box over R_6.

34	40	42	36
39	33	47	44
43	38	37	35
45	46	41	32

10.4.5 Construction of S-box through Multiplicative Group of R_7

The size of chain ring $R_7 = \frac{\mathbb{F}_2[u]/}{<u^7>} = \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2 + u^5\mathbb{F}_2 + u^6\mathbb{F}_2$ is 128. The chain of ideals is $< 0 >= u^7R_7 \subset u^6R_7 \subset u^5R_7 \subset u^4R_7 \subset u^3R_7 \subset u^2R_7 \subset uR_7 \subset R_7$. Accordingly the residue field of R_7 is $\frac{R_7}{uR_7} \simeq \mathbb{F}_2$. The ring R_7 shares some properties of the local ring $\mathbb{Z}_{2^7}$ and the Galois field $\mathbb{F}_{2^7}$. Its multiplicative part is given as follows:

$$\begin{aligned} M_{G_7} \quad = \quad & \{1, 1+u, 1+u^2, 1+u^3, 1+u^4, 1+u^5, 1+u^6, 1+u+u^2, 1+u+u^3, 1+u+u^4, \\ & 1+u+u^5, 1+u+u^6, 1+u^2+u^3, 1+u^2+u^4, 1+u^2+u^5, 1+u^2+u^6, 1+u^3+u^4, \\ & 1+u^3+u^5, 1+u^3+u^6, 1+u^4+u^5, 1+u^4+u^6, 1+u^5+u^6, 1+u+u^2+u^3, \\ & 1+u+u^2+u^4, 1+u+u^2+u^5, 1+u+u^2+u^6, 1+u^2+u^3+u^4, 1+u+u^3+u^4, \\ & 1+u+u^3+u^5, 1+u+u^3+u^6, 1+u+u^4+u^5, 1+u+u^4+u^6, 1+u+u^5+u^6, \\ & 1+u^2+u^3+u^5, 1+u^2+u^3+u^6, 1+u^2+u^4+u^5, 1+u^2+u^4+u^6, \\ & 1+u^2+u^5+u^6, 1+u^3+u^4+u^5, 1+u^3+u^4+u^6, 1+u^3+u^5+u^6, \\ & 1+u^4+u^5+u^6, 1+u+u^2+u^3+u^4, 1+u+u^2+u^3+u^5, 1+u+u^2+u^3+u^6, \\ & 1+u+u^2+u^4+u^5, 1+u+u^2+u^4+u^6, 1+u+u^2+u^5+u^6, 1+u+u^3+u^4+u^5, \\ & 1+u+u^3+u^4+u^6, 1+u+u^3+u^5+u^6, 1+u+u^4+u^5+u^6, 1+u^2+u^3+u^4+u^5, \\ & 1+u^2+u^3+u^4+u^6, 1+u^2+u^3+u^5+u^6, 1+u^2+u^4+u^5+u^6, 1+u^3+u^4+u^5+u^6, \\ & 1+u+u^2+u^3+u^4+u^5, 1+u+u^2+u^3+u^4+u^6, 1+u+u^2+u^3+u^5+u^6, \\ & 1+u+u^2+u^3+u^4+u^5+u^6, 1+u+u^3+u^4+u^5+u^6, 1+u^2+u^3+u^4+u^5+u^6, \\ & 1+u+u^2+u^3+u^4+u^5+u^6\}. \end{aligned}$$

The multiplicative subgroup M_{G_7} contains 64 elements, with 32 elements of order 8, 24 elements of order 4, 7 elements of order 2 and 1 element of order 1. Since we require the subgroups of size 16, it follows that we can fulfill our requirement by above explained availability for M_{G_7}. For this purpose we choose a subgroup $H_{G_7} = \langle 1+u^3, 1+u^2+u^3 \rangle$ of cardinality 16 of the multiplicative group M_{G_7}. Define the maps $f : H_{G_7} \to H_{G_7}$ by $f(a) = a^{-1}$ and $g : H_{G_7} \to H_{G_7}$ by $g(a) = a'a$, where $a' = 1+u^3$. Thus,

$(g \circ f)(a) = (a'a)^{-1}$. The following Table 10.19 is of $f \circ g(H_{G_7})$ in decimal form, which is in fact the S-box constructed over the chain ring $R_7 = \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2 + u^5\mathbb{F}_2 + u^6\mathbb{F}_2$.

Table. 10.19: S-box over R_7.

73	65	64	72
86	77	82	69
93	89	87	76
83	92	76	88

10.4.6 Construction of S-box through multiplicative group of R_8

The ring $R_8 = \frac{\mathbb{F}_2[u]}{<u^8>} = \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2 + u^5\mathbb{F}_2 + u^6\mathbb{F}_2 + u^7\mathbb{F}_2$ is a commutative chain ring of 2^8 elements. Since u is nilpotent with nilpotency index 8, it follows that $< 0 >= u^8R_8 \subset u^7R_8 \subset u^6R_8 \subset u^5R_8 \subset u^4R_8 \subset u^3R_8 \subset u^2R_8 \subset uR_8 \subset R_8$ and $\frac{R_8}{uR_8} \simeq \mathbb{F}_2$ is the residue field of R_8. The ring R_8 shares some properties of the local ring $\mathbb{Z}_{2^8}$ and the Galois field $\mathbb{F}_{2^8}$. The multiplication binary operation of R_8 coincides with of $\mathbb{Z}_{2^8}$, whereas the addition binary operation is similar to that of $\mathbb{F}_{2^8}$. Multiplicative group of the ring R_8 is:

$$\begin{aligned} M_{G_8} \;=\; & \{1, 1+u, 1+u^2, 1+u^3, 1+u^4, 1+u^5, 1+u^6, 1+u^7, 1+u+u^2, 1+u+u^3, \\ & 1+u+u^4, 1+u+u^5, 1+u+u^6, 1+u+u^7, 1+u^2+u^3, 1+u^2+u^4, 1+u^2+u^5, \\ & 1+u^2+u^6, 1+u^2+u^7, 1+u^3+u^4, 1+u^3+u^5, 1+u^3+u^6, 1+u^3+u^7, 1+u^4+u^5, \\ & 1+u^4+u^6, 1+u^4+u^7, 1+u^5+u^6, 1+u^5+u^7, 1+u^6+u^7, 1+u+u^2+u^3, \\ & 1+u+u^2+u^4, 1+u+u^2+u^5, 1+u+u^2+u^6, 1+u+u^2+u^7, 1+u+u^3+u^4, \\ & 1+u+u^3+u^5, 1+u+u^3+u^6, 1+u+u^3+u^7, 1+u+u^4+u^5, 1+u+u^4+u^6, \\ & 1+u+u^4+u^7, 1+u+u^5+u^6, 1+u+u^5+u^7, 1+u+u^6+u^7, 1+u^2+u^3+u^4, \\ & 1+u^2+u^3+u^5, 1+u^2+u^3+u^6, 1+u^2+u^3+u^7, 1+u^2+u^4+u^5, 1+u^2+u^4+u^6, \\ & 1+u^2+u^4+u^7, 1+u^2+u^5+u^6, 1+u^2+u^5+u^7, 1+u^2+u^6+u^7, 1+u^3+u^4+u^5, \\ & 1+u^3+u^4+u^6, 1+u^3+u^4+u^7, 1+u^3+u^5+u^6, 1+u^3+u^5+u^7, 1+u^3+u^6+u^7, \\ & 1+u^4+u^5+u^6, 1+u^4+u^5+u^7, 1+u^4+u^6+u^7, 1+u^5+u^6+u^7, 1+u+u^2+u^3+u^4, \\ & 1+u+u^2+u^3+u^5, 1+u+u^2+u^3+u^6, 1+u+u^2+u^3+u^7, 1+u+u^2+u^4+u^5, \\ & 1+u+u^2+u^4+u^6, 1+u+u^2+u^4+u^7, 1+u+u^2+u^5+u^6, 1+u+u^2+u^5+u^7, \\ & 1+u+u^2+u^6+u^7, 1+u+u^3+u^4+u^5, 1+u+u^3+u^4+u^6, 1+u+u^3+u^4+u^7, \\ & 1+u+u^3+u^5+u^6, 1+u+u^3+u^5+u^7, 1+u+u^3+u^6+u^7, 1+u+u^4+u^5+u^6, \\ & 1+u+u^4+u^5+u^7, 1+u+u^4+u^6+u^7, 1+u+u^5+u^6+u^7, 1+u^2+u^3+u^4+u^5, \\ & 1+u^2+u^3+u^4+u^6, 1+u^2+u^3+u^4+u^7, 1+u^2+u^3+u^5+u^6, 1+u^2+u^3+u^5+u^7, \end{aligned}$$

$1+u^2+u^3+u^6+u^7, 1+u^2+u^4+u^5+u^6, 1+u^2+u^4+u^5+u^7, 1+u^2+u^4+u^6+u^7,$

$1+u^2+u^5+u^6+u^7, 1+u^3+u^4+u^5+u^6, 1+u^3+u^4+u^5+u^7, 1+u^3+u^4+u^6+u^7,$

$1+u^3+u^5+u^6+u^7, 1+u^4+u^5+u^6+u^7, 1+u+u^2+u^3+u^4+u^5, 1+u+u^2+u^3+u^4$

$+u^6,\ 1+u+u^2+u^3+u^5+u^6, 1+u+u^2+u^4+u^5+u^6, 1+u+u^3+u^4+u^5+u^6,$

$1+u^2+u^3+u^4+u^5+u^6, 1+u+u^2+u^3+u^4+u^7, 1+u+u^2+u^3+u^5+u^7,$

$1+u+u^2+u^4+u^5+u^7, 1+u+u^3+u^4+u^5+u^7, 1+u^2+u^3+u^4+u^5+u^7,$

$1+u+u^2+u^3+u^6+u^7, 1+u+u^2+u^4+u^6+u^7, 1+u+u^3+u^4+u^6+u^7,$

$1+u^2+u^3+u^4+u^6+u^7, 1+u+u^2+u^5+u^6+u^7, 1+u+u^3+u^5+u^6+u^7,$

$1+u^2+u^3+u^5+u^6+u^7, 1+u+u^4+u^5+u^6+u^7, 1+u^2+u^4+u^5+u^6+u^7,$

$1+u^3+u^4+u^5+u^6+u^7 1+u+u^2+u^3+u^4+u^5+u^6, 1+u+u^2+u^3+u^4+u^5+u^7,$

$1+u+u^2+u^3+u^4+u^6+u^7, 1+u+u^2+u^3+u^5+u^6+u^7, 1+u+u^2+u^4+u^5+u^6+u^7,$

$1+u+u^3+u^4+u^5+u^6+u^7, 1+u^2+u^3+u^4+u^5+u^6+u^7,$

$1+u+u^2+u^3+u^4+u^5+u^6+u^7\}.$

The multiplicative group M_{G_8} contains 128 elements, 64 elements of order 8, 48 elements of order 4, 15 elements of order 2 and 1 element of order 1. Ever since we require the subgroups of cardinality 16, therefore we accomplish our constraint by above explained availability for M_{G_6}, and set of all elements of order 2 also generate a subgroup of order 16. We choose a subgroup $H_{G_8} = \langle 1+u^3+u^6, 1+u^2+u^4+u^5+u^7 \rangle$ of the group M_{G_8} having cardinality 16. Define the maps $f : H_{G_8} \to H_{G_8}$ by $f(a) = a^{-1}$ and $g : H_{G_8} \to H_{G_8}$ by $g(a) = a'a$, where we take $a' = 1+u^4+u^6$. Thus, $(g \circ f)(a) = (a'a)^{-1}$. The following Table 10.20 is of $f \circ g(H_{G_8})$ in decimal form, which is in fact the S-box designed over the chain ring $R_8 = \mathbb{F}_2 + u\mathbb{F}_2 + +u^2\mathbb{F}_2 + u^3\mathbb{F}_2 + u^4\mathbb{F}_2 + u^5\mathbb{F}_2 + u^6\mathbb{F}_2 + u^7\mathbb{F}_2$.

Table 10.20: S-box over R_8.

138	153	130	136
155	175	165	186
146	177	128	173
167	184	143	179

10.5 Applications of Proposed Substitution Box in Image Encryption and Watermarking

As digital image plays an important role in multimedia technology, it becomes more important for the user's to maintain privacy. And to provide such security and privacy to the user, encryption and watermarking is very important to protect from any unauthorized user access. The encryption and watermarking have applications in various fields, including internet communication, multimedia systems, medical imaging, Telemedicine and military communication. Nowadays, the prominent share of the multimedia fabrication and dissemination is carried out digitally. The rapid growth of digital media like Internet and Compact Discs has ushered in a wonderful era where the flow, duplication and modification of digital images have become all the more easier and simpler. Mega distribution of flawless replicas of multimedia data at an accelerated degree has become the order of the day. And this phenomenon has unfortunately resulted in tremendous threats to multimedia safety and copyright security. This has the effect of ringing an alarm bell for authors, when the stark reality dawned upon them, convincing that conservative safety systems, like encryption were incapable of affording the much-needed shelter. This has motivated many investigators to devise alternate methods, one of which is known by the term 'digital watermarking' which is nothing but the art of concealing data in a healthy way and without being noticed by pirates or others of the sort. The classifications of information hiding techniques are cryptography, watermarking and steganography. Here we will only focus on encryption that belongs to cryptography and watermarking. Encryption protects content during the transmission of the data from the sender to receiver. However, after receipt and subsequent decoding, the data is no longer protected and is in the clear. Watermarking compliments encryption by embedding a signal directly into the data. Thus, the goal of a watermarking is to always remain present in the data. The algorithms for image encryption and watermarking schemes are presented in Figs. 10.1-10.2.

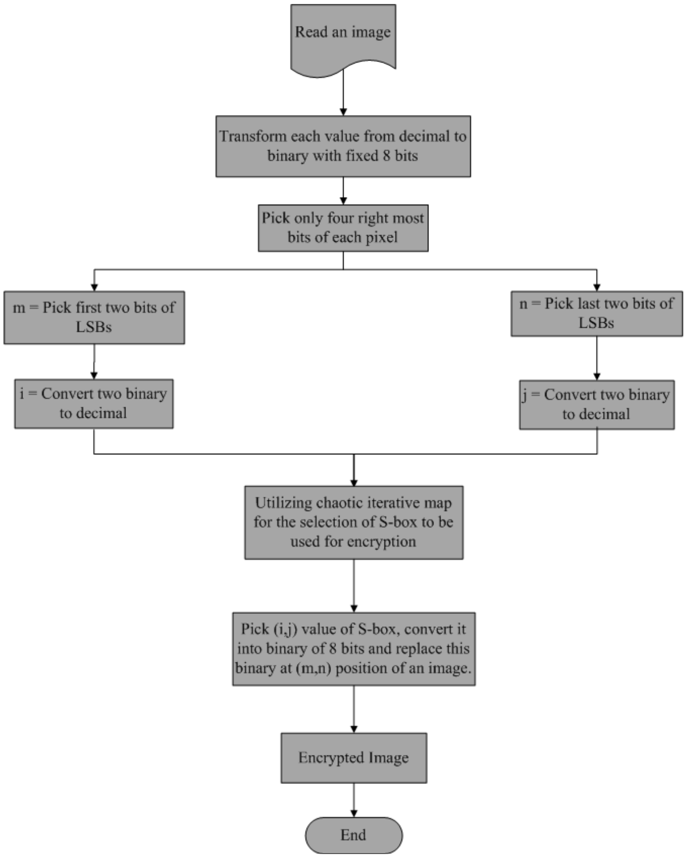

Read an image
Transform each value from decimal to binary with fixed 8 bits
Pick only four right most bits of each pixel
m = Pick first two bits of LSBs
n = Pick last two bits of LSBs
i = Convert two binary to decimal
j = Convert two binary to decimal
Utilizing chaotic iterative map for the selection of S-box to be used for encryption
Pick (i,j) value of S-box, convert it into binary of 8 bits and replace this binary at (m,n) position of an image.
Encrypted Image
End

Fig. 10.1: Proposed image encryption algorithm based on Galois ring.

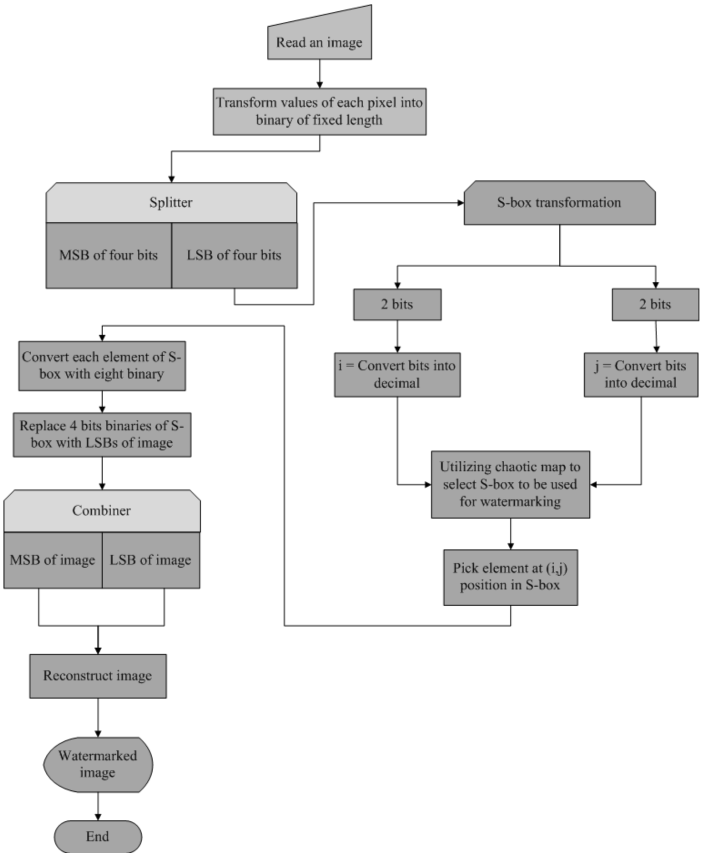

Read an image
Transform values of each pixel into binary of fixed length
Splitter
MSB of four bits
LSB of four bits
S-box transformation
2 bits
2 bits
i = Convert bits into decimal
j = Convert bits into decimal
Utilizing chaotic map to select S-box to be used for watermarking
Pick element at (i,j) position in S-box
Convert each element of S-box with eight binary
Replace 4 bits binaries of S-box with LSBs of image
Combiner
MSB of image
LSB of image
Reconstruct image
Watermarked image
End

Fig. 10.2: Algorithm for image watermarking based on Galois ring.

The results after applying the proposed image encryption and watermarking schemes are given in Figs. 10.3-10.6 respectively.

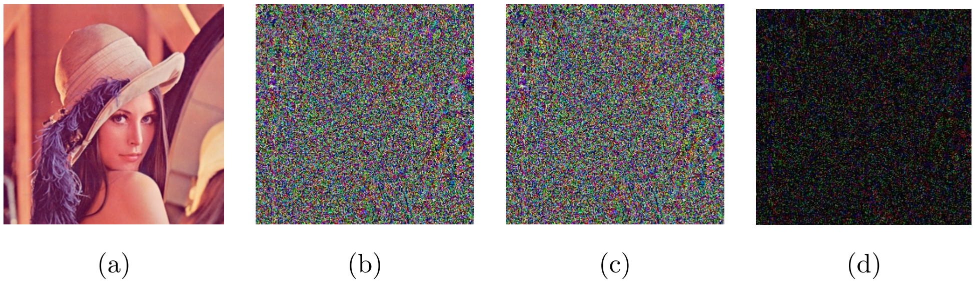

(a) (b) (c) (d)

Fig. 10.3: (a) Plain Lena image, (b) Encrypted image using GR(4,4), (c) Encrypted image using GR(8,4), (d) Encrypted image using GR(32,4).

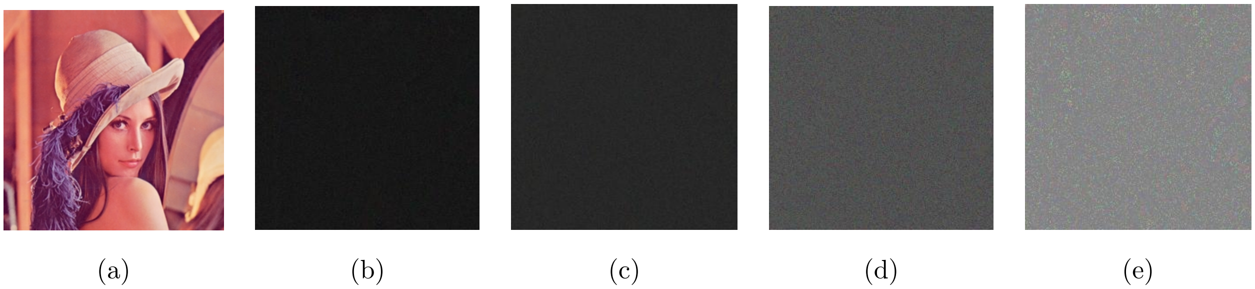


Fig. 10.4: (a) Plain Lena image, (b) Encrypted image using R_5, (c) Encrypted image using R_6, (d) Encrypted image using R_7, (e) Encrypted image using R_8.

Fig. 10.5: (a) Cover Lena image, (b) Watermarked image using $GR(4,4)$, (c) Watermarked image using $GR(8,4)$, (d) Watermarked image using $GR(32,4)$.

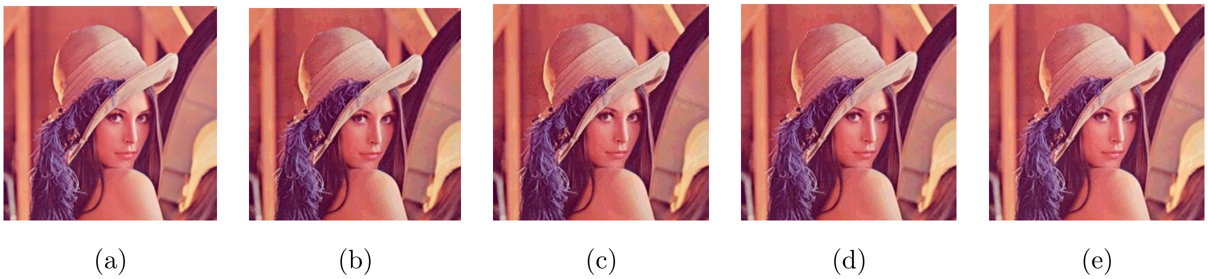


Fig. 10.6: (a) Cover Lena image, (b) Watermarked image using R_5, (c) Watermarked image using R_6, (d) Watermarked image using R_7, (e) Watermarked image using R_8.

The statistical analyses plays an important role in estimating good quality information hiding. We have applies first order texture image analysis that deals with the histograms of an image which includes mean, standard deviation (Std.), skewness and kurtosis. The GLCM analyses of an image consists of entropy, contrast, homogeneity, energy and correlation. The correlation based statistical anlyses consists of structure content, normalized cross correlation. The human visual system (HVS) fundamentally deals with the human perceptions. These analyses includes universal image quality index, structure content and structure similarity index metric.

Table 10.21: First order texture analysis of proposed encryption scheme based on S-box of $GR(4,4)$.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.300781	0.355469	0.292969
Std.	0.496541	0.459496	0.38138	0.459496	0.479593	0.456016
Skewness	-0.267999	0.868817	1.703557	0.868817	0.603906	0.909779
Kurtosis	1.07182	1.75484	3.90216	1.754840	1.364700	1.827700

Table 10.22: First order texture analysis of proposed encryption scheme based on S-box of $GR(8,4)$.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.261719	0.210938	0.183594
Std.	0.496541	0.459496	0.38138	0.440431	0.408773	0.387911
Skewness	-0.267999	0.868817	1.703557	1.084160	1.417060	1.634530
Kurtosis	1.07182	1.75484	3.90216	2.173900	3.008070	3.671690

Table 10.23: First order texture analysis of proposed encryption scheme based on S-box of $GR(32,4)$.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.101563	0.136719	0.078125
Std.	0.496541	0.459496	0.38138	0.302664	0.344223	0.268894
Skewness	-0.267999	0.868817	1.703557	2.638030	2.114870	3.144000
Kurtosis	1.07182	1.75484	3.90216	7.959200	5.472660	10.884700

Table 10.24: First order texture analysis of proposed encryption scheme based on S-box of R_5.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.578125	0.5156250	0.382813
Std.	0.496541	0.459496	0.381380	0.494826	0.5007350	0.487025
Skewness	-0.267999	0.8688817	1.70357	-0.316386	-0.0625305	0.482181
Kurtosis	1.071820	1.754840	3.90216	1.100100	1.0039100	1.232500

Table 10.25: First order texture analysis of proposed encryption scheme based on S-box of R_6.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.628906	0.636719	0.597656
Std.	0.496541	0.459496	0.381380	0.484044	0.481887	0.491331
Skewness	-0.267999	0.8688817	1.70357	-0.533666	-0.568542	-0.398296
Kurtosis	1.071820	1.754840	3.90216	1.284800	1.323240	1.15864

Table. 10.26: First order texture analysis of proposed encryption scheme based on S-box of R_7.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.421875	0.613281	0.558594
Std.	0.496541	0.459496	0.38138	0.496541	0.487952	0.497528
Skewness	-0.267999	0.868817	1.703557	0.316386	-0.465222	-0.236001
Kurtosis	1.07182	1.75484	3.90216	1.100100	1.216430	1.055700

Table 10.27: First order texture analysis of proposed encryption scheme based on S-box of R_8.

	Plain image color components			Encrypted image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.5090600	0.4648440	0.4960940
Std.	0.496541	0.459496	0.38138	0.5009640	0.4997400	0.5009640
Skewness	-0.267999	0.868817	1.703557	-0.0156255	0.1400974	0.0156255
Kurtosis	1.07182	1.75484	3.90216	1.0002400	1.0198700	1.0002400

Table 10.28: Second order texture analysis of proposed encryption scheme based on S-box of $GR(4,4)$.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES [108]
Contrast	0.372687	0.392816	0.365273	5.716670	5.930760	5.653100	5.766843	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.463198	0.460000	0.463737	0.462311	0.4701
Entropy	7.2911	7.58133	7.07945	7.724020	7.743380	7.694770	7.720723	7.9325
Correlation	0.923453	0.929416	0.853838	0.0796321	0.0854177	0.0696044	0.078218	0.0815
Energy	0.138624	0.0999494	0.169877	0.0247005	0.0242131	0.0250313	0.024648	0.0211

Table 10.29: Second order texture analysis of proposed encryption scheme based on S-box of $GR(8,4)$.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES [108]
Contrast	0.372687	0.392816	0.365273	7.344910	7.551030	7.294580	7.396840	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.523815	0.524585	0.521437	0.523279	0.4701
Entropy	7.29110	7.581330	7.079450	7.513200	7.738900	7.099640	7.450580	7.9325
Correlation	0.923453	0.929416	0.853838	0.0394828	0.0450918	0.0250336	0.036536	0.0815
Energy	0.138624	0.0999494	0.169877	0.0536281	0.0600180	0.0509045	0.054848	0.0211

Table 10.30: Second order texture analysis of proposed encryption scheme based on S-box of $GR(32,4)$.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES [108]
Contrast	0.372687	0.392816	0.365273	7.091800	7.103510	7.079500	7.09160	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.742998	0.754889	0.741075	0.74321	0.4701
Entropy	7.29110	7.581330	7.079450	7.416100	7.513900	7.10156	7.34385	7.9325
Correlation	0.923453	0.929416	0.853838	0.026632	0.025921	0.016139	0.02289	0.0815
Energy	0.138624	0.0999494	0.169877	0.283568	0.320313	0.278132	0.29400	0.0211

Table 10.31: Second order texture analysis of proposed encryption scheme based on S-box of R_5.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES [108]
Contrast	0.372687	0.392816	0.365273	7.002053	7.0009038	7.002497	7.001818	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.468974	0.469548	0.468752	0.469095	0.4701
Entropy	7.29110	7.581330	7.079450	7.354130	7.709100	7.099640	7.386680	7.9325
Correlation	0.923453	0.929416	0.853838	0.191800	0.0323392	0.27433	0.166156	0.0815
Energy	0.138624	0.0999494	0.169877	0.025412	0.028163	0.024069	0.025881	0.0211

Table 10.32: Second order texture analysis of proposed encryption scheme based on S-box of R_6.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES[108]
Contrast	0.372687	0.392816	0.365273	7.0006433	7.000567	7.000781	7.000663	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.4696778	0.479717	0.459609	0.469668	0.4701
Entropy	7.29110	7.581330	7.079450	7.4561870	7.7813561	7.351237	7.529593	7.9325
Correlation	0.923453	0.929416	0.853838	-0.000321795	0.0510155	0.037352	0.029348	0.0815
Energy	0.138624	0.0999494	0.169877	0.0198714	0.018836	0.021408	0.020038	0.0211

Table 10.33: Second order texture analysis of proposed encryption scheme based on S-box of R_7.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES[108]
Contrast	0.372687	0.392816	0.365273	7.010233	7.008762	7.010815	7.00993	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.484924	0.485660	0.484654	0.48508	0.4701
Entropy	7.29110	7.581330	7.079450	7.513200	7.738900	7.099640	7.45058	7.9325
Correlation	0.923453	0.929416	0.853838	0.022019	0.009103	0.0337697	0.02163	0.0815
Energy	0.138624	0.0999494	0.169877	0.02550	0.023261	0.0208312	0.02400	0.0211

Table 10.34: Second order texture analysis of proposed encryption scheme based on S-box of R_8.

	Plain image color components			Encrypted image color components				
	Red	Green	Blue	Red	Green	Blue	Average	AES [108]
Contrast	0.372687	0.392816	0.365273	7.620619	7.602711	7.619807	7.614379	7.2240
Homogeneity	0.872453	0.871262	0.874949	0.439325	0.452688	0.478967	0.456993	0.4701
Entropy	7.29110	7.581330	7.079450	7.04684	7.02036	7.04413	7.037110	7.9325
Correlation	0.923453	0.929416	0.853838	0.0572997	0.0437702	0.0580387	0.053036	0.0815
Energy	0.138624	0.0999494	0.169877	0.020234	0.025695	0.020018	0.021982	0.0211

Table 10.35: Image error measurements of proposed encryption scheme based on S-box of $GR(4, 4)$.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	12134.3	6068.13	4437.92				
Peak Signal to Noise Ratio	7.29067	10.3003	11.6590	9.74999	8.1421	9.0014	9.2541
Mean Absolute Error	93.3373	63.2998	54.0589				

Table 10.36: Image error measurements of proposed encryption scheme based on S-box of $GR(8, 4)$.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	19007.1	6839.19	5523.35	-			
Peak Signal to Noise Ratio	5.37564	9.81475	10.7428	8.64439	8.1421	9.0014	9.2541
Mean Absolute Error	122.514	67.7453	61.6997				

Table 10.37: Image error measurements of proposed encryption scheme based on S-box of $GR(32, 4)$.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	26869.2	8949.01	8245.61	-			
Peak Signal to Noise Ratio	8.83825	8.61305	8.96858	7.13996	8.1421	9.0014	9.2541
Mean Absolute Error	154.441	79.2899	80.9500				

Table 10.38: Image error measurements of proposed encryption scheme based on S-box of R_5.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	26938.7	8395.85	7914.18	-			
Peak Signal to Noise Ratio	8.62704	8.89119	9.14675	8.80662	8.1421	9.0014	9.2541
Mean Absolute Error	156.614	76.4777	81.7981				

Table 10.39: Image error measurements of proposed encryption scheme based on S-box of R_6.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	22211.3	6234.6	5573.56	-			
Peak Signal to Noise Ratio	4.66506	10.1827	10.6695	8.50575	8.1421	9.0014	9.2541
Mean Absolute Error	140.6860	64.5333	66.0316				

Table 10.40: Image error measurments of proposed encryption scheme based on S-box of R_7.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	12225.6	3037.64	1838.02	-			
Peak Signal to Noise Ratio	7.2581	13.3054	15.4873	12.0169	8.1421	9.0014	9.2541
Mean Absolute Error	99.2633	45.0841	32.8454				

Table 10.41: Image error measurments of proposed encryption scheme based on S-box of R_8.

	Image color components						
	Red	Green	Blue	Average	Gray [108]	APA [108]	Lui [108]
Mean Square Error	3269.08	6173.97	4007.07	-			
Peak Signal to Noise Ratio	12.9866	10.2252	12.1025	11.7814	8.1421	9.0014	9.2541
Mean Absolute Error	50.4345	66.0888	55.0974				

Table 10.42: Image similarity measurments of proposed encryption scheme based on S-box of $GR(4,4)$.

	Image color components		
	Red	Green	Blue
Structure Content	2.67522000	0.959737000	0.9467440
Universal Image Quality Index	-0.00329472	0.000386892	-0.0000435
Structure Similarity Index Metric	0.013055400	0.016328500	0.0184890

Table 10.43: Image similarity measurments of proposed encryption scheme based on S-box of $GR(8,4)$.

	Image color components		
	Red	Green	Blue
Structure Content	5.5605800	2.01720000	1.96540000
Universal Image Quality Index	-0.0016198	-0.00399473	-0.00170602
Structure Similarity Index Metric	0.0130455	0.01506070	0.019223800

Table 10.44: Image similarity measurments of proposed encryption scheme based on S-box of $GR(32,4)$.

	Image color components		
	Red	Green	Blue
Structure Content	25.7971000	9.660140000	9.28556
Universal Image Quality Index	0.000360266	0.000617253	-0.000210373
Structure Similarity Index Metric	0.021377500	0.037992700	0.036532900

Table 10.45: Image similarity measurments of proposed encryption scheme based on S-box of R_5.

	Image color components		
	Red	Green	Blue
Structure Content	62.221	22.4185	21.9321
Universal Image Quality Index	-0.00113042	-0.0000975	-0.00306773
Structure Similarity Index Metric	0.119023	0.219966	0.212875

Table 10.46: Image similarity measurments of proposed encryption scheme based on S-box of R_6.

	Image color components		
	Red	Green	Blue
Structure Content	5.5605800	2.01720000	1.96540000
Universal Image Quality Index	-0.0016198	-0.00399473	-0.00170602
Structure Similarity Index Metric	0.0130455	0.01506070	0.019223800

Table 10.47: Image similarity measurments of proposed encryption scheme based on S-box of R_7.

	Image color components		
	Red	Green	Blue
Structure Content	22.3112000	7.980790000	7.90325000
Universal Image Quality Index	0.00290522	0.000516327	0.00219534
Structure Similarity Index Metric	0.19237900	0.301132000	0.3223110

Table 10.48: Image similarity measurments of proposed encryption scheme based on S-box of R_8.

	Image color components		
	Red	Green	Blue
Structure Content	5.2635300	1.8889400	1.86378000
Universal Image Quality Index	0.0023029	-0.0020651	-0.0029209
Structure Similarity Index Metric	0.2721280	0.3122190	0.36223600

Table 10.49: First order texture analysis of proposed watermarking scheme based on S-box of $GR(4,4)$.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.574219	0.296875	0.195313
Std.	0.496541	0.459496	0.38138	0.495429	0.457776	0.397218
Skewness	-0.267999	0.868817	1.70357	-0.300201	0.889181	1.53711
Kurtosis	1.07182	1.75484	3.90216	1.09012	1.79064	3.36272

Table 10.50: First order texture analysis of proposed watermarking scheme based on S-box of $GR(8,4)$.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.589844	0.31250	0.160156
Std.	0.496541	0.459496	0.38138	0.492825	0.46442	0.367469
Skewness	-0.267999	0.868817	1.70357	-0.365321	0.80904	1.853270
Kurtosis	1.07182	1.75484	3.90216	1.13346	1.65455	4.434600

Table 10.51: First order texture analysis of proposed watermarking scheme based on S-box of $GR(32,4)$.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.589844	0.3125	0.160156
Std.	0.496541	0.459496	0.38138	0.492825	0.46442	0.367469
Skewness	-0.267999	0.868817	1.70357	-0.365321	0.80904	1.85327
Kurtosis	1.07182	1.75484	3.90216	1.13346	1.65455	4.4346

Table 10.52: First order texture analysis of proposed watermarking scheme based on S-box of R_5.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.574219	0.28125	0.175781
Std.	0.496541	0.459496	0.38138	0.495429	0.45049	0.38138
Skewness	-0.267999	0.868817	1.703557	-0.300201	0.973067	1.70357
Kurtosis	1.07182	1.75484	3.90216	1.09012	1.94686	3.90216

Table 10.53: First order texture analysis of proposed watermarking scheme based on S-box of R_6.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.574219	0.28125	0.175781
Std.	0.496541	0.459496	0.38138	0.495429	0.45049	0.38138
Skewness	-0.267999	0.868817	1.703557	-0.300201	0.973067	1.70357
Kurtosis	1.07182	1.75484	3.90216	1.09012	1.94686	3.90216

Table 10.54: First order texture analysis of proposed watermarking scheme based on S-box of R_7.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.578125	0.289063	0.183594
Std.	0.496541	0.459496	0.38138	0.494826	0.454215	0.387911
Skewness	-0.267999	0.868817	1.703557	-0.316386	0.930620	1.634530
Kurtosis	1.07182	1.75484	3.90216	1.100100	1.866050	3.671690

Table 10.55: First order texture analysis of proposed watermarking scheme based on S-box of R_8.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.570313	0.296875	0.171875
Std.	0.496541	0.459496	0.38138	0.496001	0.457776	0.378011
Skewness	-0.267999	0.868817	1.703557	-0.284073	0.889181	1.739460
Kurtosis	1.07182	1.75484	3.90216	1.080700	1.790640	4.025730

Table 10.56: Second order texture analysis of proposed watermarking scheme based on S-box of $GR(4,4)$.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.39375	0.406985	0.389338
Homogeneity	0.872453	0.871262	0.874949	0.864794	0.866005	0.865558
Entropy	7.29110	7.581330	7.079450	7.32279	7.56524	7.09129
Correlation	0.923453	0.929416	0.853838	0.920109	0.926875	0.847282
Energy	0.138624	0.0999494	0.169877	0.135096	0.0973498	0.159161

Table 10.57: Second order texture analysis of proposed watermarking scheme based on S-box of $GR(8,4)$.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.391866	0.406127	00387469
Homogeneity	0.872453	0.871262	0.874949	0.865176	0.866715	0.868536
Entropy	7.29110	7.581330	7.079450	7.3227	7.5607	7.08971
Correlation	0.923453	0.929416	0.853838	0.920656	0.927186	0.846354
Energy	0.138624	0.0999494	0.169877	0.134363	0.0978212	0.161773

Table 10.58: Second order texture analysis of proposed watermarking scheme based on S-box of $GR(32,4)$.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.391866	0.406127	0.387469
Homogeneity	0.872453	0.871262	0.874949	0.865176	0.866715	0.868536
Entropy	7.29110	7.581330	7.079450	7.3227	7.5607	7.08971
Correlation	0.923453	0.929416	0.853838	0.920656	0.927186	0.846354
Energy	0.138624	0.0999494	0.169877	0.134363	0.0978212	0.161773

Table 10.59: Second order texture analysis of proposed watermarking scheme based on S-box of R_5.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.397702	0.403278	0.376716
Homogeneity	0.872453	0.871262	0.874949	0.863205	0.863345	0.870555
Entropy	7.2911	7.58133	7.07945	7.30967	7.48019	7.0773
Correlation	0.923453	0.929416	0.853838	0.92072	0.926736	0.854595
Energy	0.138624	0.0999494	0.169877	0.133659	0.0969861	0.155589

Table 10.60: Second order texture analysis of proposed watermarking scheme based on S-box of R_6.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.394225	0.395787	0.374494
Homogeneity	0.872453	0.871262	0.874949	0.864685	0.867443	0.870939
Entropy	7.29110	7.581330	7.079450	7.30751	7.48453	7.07733
Correlation	0.923453	0.929416	0.853838	0.921642	0.928676	0.854764
Energy	0.138624	0.0999494	0.169877	0.135017	0.098445	0.157312

Table 10.61: Second order texture analysis of proposed watermarking scheme based on S-box of R_7.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.382154	0.383824	0.373851
Homogeneity	0.872453	0.871262	0.874949	0.869645	0.874147	0.872720
Entropy	7.29110	7.581330	7.079450	7.29890	7.499040	7.077940
Correlation	0.923453	0.929416	0.853838	0.923301	0.931268	0.853531
Energy	0.138624	0.0999494	0.169877	0.137470	0.100933	0.161723

Table 10.62: Second order texture analysis of proposed watermarking scheme based on S-box of R_8.

	Original image color components			Watermarked image color components		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.376532	0.399203	0.374295
Homogeneity	0.872453	0.871262	0.874949	0.871378	0.868688	0.871490
Entropy	7.29110	7.581330	7.079450	7.282660	7.512610	7.076750
Correlation	0.923453	0.929416	0.853838	0.921681	0.928119	0.850357
Energy	0.138624	0.0999494	0.169877	0.140061	0.098777	0.168454

Table 10.63: Image error measurements of proposed watermarking scheme based on S-box of $GR(4,4)$.

	Image color components		
	Red	Green	Blue
Mean Square Error	35.072	30.841	37.9247
Peak Signal to Noise Ratio	32.6812	33.2395	32.3416
Mean Absolute Error	4.62018	4.33269	4.80144

Table 10.64: Image error measurements of proposed watermarking scheme based on S-box of $GR(8,4)$.

	Image color components		
	Red	Green	Blue
Mean Square Error	29.9243	26.5959	32.3507
Peak Signal to Noise Ratio	33.3706	33.8827	33.0320
Mean Absolute Error	4.22232	3.98900	4.40581

Table 10.65: Image error measurements of proposed watermarking scheme based on S-box of $GR(32,4)$.

	Image color components		
	Red	Green	Blue
Mean Square Error	29.9243	26.5959	32.3507
Peak Signal to Noise Ratio	33.3706	33.8827	33.0320
Mean Absolute Error	4.22232	3.98900	4.40581

Table 10.66: Image error measurments of proposed watermarking scheme based on S-box of R_5.

	Image color components		
	Red	Green	Blue
Mean Square Error	67.7634	60.3825	67.1817
Peak Signal to Noise Ratio	29.8206	30.3217	29.8583
Mean Absolute Error	7.09592	6.75697	6.99326

Table 10.67: Image error measurments of proposed watermarking scheme based on S-box of R_6.

	Image color components		
	Red	Green	Blue
Mean Square Error	55.2887	48.5003	55.1235
Peak Signal to Noise Ratio	30.7044	31.2734	30.7174
Mean Absolute Error	6.26427	5.9082	6.19368

Table 10.68: Image error measurments of proposed watermarking scheme based on S-box of R_7.

	Image color components		
	Red	Green	Blue
Mean Square Error	31.9533	26.3285	32.6609
Peak Signal to Noise Ratio	33.0856	33.9265	32.9905
Mean Absolute Error	5.65273	5.13113	5.71497

Table 10.69: Image error measurments of proposed watermarking scheme based on S-box of R_8.

	Image color components		
	Red	Green	Blue
Mean Square Error	22.435	20.2259	24.6088
Peak Signal to Noise Ratio	34.6215	35.0717	34.2199
Mean Absolute Error	3.65494	3.48647	3.83476

Table 10.70: Image similarity measurments of proposed watermarking scheme based on S-box of $GR(4,4)$.

	Image color components		
	Red	Green	Blue
Structure Content	1.02006	1.02876	1.03460
Universal Image Quality Index	0.767415	0.80177	0.756734
Structure Similarity Index Metric	0.895856	0.906332	0.885709

Table 10.71: Image similarity measurments of proposed watermarking scheme based on S-box of $GR(8,4)$.

	Image color components		
	Red	Green	Blue
Structure Content	1.01492	1.02385	1.02670
Universal Image Quality Index	0.78181	0.812344	0.767011
Structure Similarity Index Metric	0.906944	0.917087	0.896842

Table 10.72: Image similarity measurments of proposed watermarking scheme based on S-box of $GR(32,4)$.

	Image color components		
	Red	Green	Blue
Structure Content	1.01492	1.023850	1.02670
Universal Image Quality Index	0.78181	0.812344	0.767011
Structure Similarity Index Metric	0.906944	0.917087	0.896842

Table 10.73: Image similarity measurments of proposed watermarking scheme based on S-box of R_5.

	Image color components		
	Red	Green	Blue
Structure Content	1.0733	1.11063	1.12045
Universal Image Quality Index	0.80846	0.837688	0.792751
Structure Similarity Index Metric	0.927252	0.932159	0.916876

Table 10.74: Image similarity measurments of proposed watermarking scheme based on S-box of R_6.

	Image color components		
	Red	Green	Blue
Structure Content	1.06197	1.09256	1.10057
Universal Image Quality Index	0.808703	0.839345	0.79333
Structure Similarity Index Metric	0.927525	0.934201	0.917581

Table 10.75: Image similarity measurments of proposed watermarking scheme based on S-box of R_7.

	Image color components		
	Red	Green	Blue
Structure Content	1.03484	1.04849	1.05593
Universal Image Quality Index	0.81822	0.85175	0.80365
Structure Similarity Index Metric	0.93213	0.94117	0.92368

Table 10.76: Image similarity measurments of proposed watermarking scheme based on S-box of R_8.

	Image color components		
	Red	Green	Blue
Structure Content	0.983519	0.969268	0.970146
Universal Image Quality Index	0.825920	0.858906	0.812379
Structure Similarity Index Metric	0.936415	0.945399	0.927879

First-order statistics are quite straightforward. They are computed from a function that measures the probability of a certain pixel occurring in an image. The interpretations of first order texture analysis of an image are quite straightforward. They are computed from the mechanism which measures the pixel probabilities in an image. The analysis of first order textures like mean, standard deviation, skewness and kurtosis reflects that there are significant changes in these features for plain and encrypted images in case of Galois rings and finite chain rings (see Tables 10.21-10.27) whereas in the case of watermarking these parameter values will remain constant with some minute changes for original and watermarked images (see Tables 10.49-10.55).

The second order texture analysis generally deals with contrast, homogeneity, entropy, correlation and energy. The contrast measures the amount of local variations present in the image. Contrast is zero when the neighboring pixels have constant values. The values of second order characteristics for plain and encrypted images are different from each other and for watermarking through Galois rings and finite chain rings are remain same or tend to cover image second order texture features (see Tables 10.28-10.34, 10.56-10.66).

The image error measurements and image similarity analysis in case of image encryption and watermarking are quite different. The values of the means square error and mean absolute error increases, whereas peak signal to noise ratio decreases for image encryption. As far as watermarking is concerned, these analyses are entirely changed. The value of mean square error and mean absolute error decreases, and peak signal to noise ratio decreases (see Tables 10.35-10.41,10.63-10.69).

The structural similarity image quality standard is grounded on the notion that the human visual system is extremely modified for extracting structural information from the scene, and therefore a measure of structural similarity can provide a good approximation to perceived image quality. The standard similarity measurement tests which include structure content, universal image quality index and structure similarity index metric (SSIM). The similarity coefficients values for image encryption and watermarking are computed (see Tables 10.42-10.48,10.70-10.76). The readings of similarity measures discloses the quality of encryption using proposed algorithms for image encryption, which is based on chain rings. The structure content values in case of image encryption are higher than unity which reveals that two images are completely different. Similarly, structure similarity index and universal image quality index measure far away from unity backwardly which guarantee the authentication of the proposed image encryption algorithm. In case of watermarking similarity coefficients are closed to one which elucidates the robustness of suggested watermarking algorithm constructed on the classes of chain rings.

10.6 Conclusion

In this chapter, we developed new schemes for image encryption and watermarking independently that soundly depends on classes of finite chain rings. The readings of test images in case of encryption and watermarking are closed to optimal values that reflect the endorsement of our suggested data hiding technique. In future, we will combine encryption and watermarking due to the fact that cryptography provides no protection once the content is decrypted, which is required for human perception, whereas watermarking complements cryptography by embedding a message within the content.

Chapter 11

Utilizing Small S-boxes in Steganography

Digital Steganography exploits the use of a cover data to hide secret information in such a way that it is imperceptible to a human observer. The secret information can be concealed in content such as image, audio, or video. This part of thesis provides a novel image steganographic technique to hide color secret image in color cover image using small S-boxes based on multiplicative group of nonzero elements of Galois field of order 16 i-e., $\mathbb{Z}_{17}^*$, symmetry group S_4 and least significant bits (LSBs). The combination of these three methods will enhance the security of the data embedded. This combined technique will fulfill the necessities such as capacity, security and robustness for secure information transmission over an open channel. A comparative scrutiny is made to show the viability of the proposed technique by first and second order texture analysis, mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), average difference (AD), normalized absolute error (NAE), maximum difference (MD), enhancement error (EME), peak signal to noise ratio (PSNR), structure contents (SC), normalized cross-correlation, universal image quality index (UIQI) and structural similarity index metric (SSIM). We investigated the information concealing strategy utilizing the picture execution parameters like first order and second order texture characteristics. The stego pictures are tried by transmitting them and the implanted information are effectively extricated by the collector. There is no visual modification between the stego image and the cover image. The investigations exhibited the high hiddenness of the suggested model even with large size secret image.

11.1 Exponential transformation

Let us consider the following function $f : M \to M$ defined as:

$$x \mapsto \begin{cases} a^m \bmod 17, & if\ x < 16, \\ 0, & if\ \ x = 16, \end{cases} \tag{11.1}$$

where $x = a^m \pmod{17}$ and $m \in M = \{0, 1, 2, ..., 15\}$. We select a as a primitive element which generates the multiplicative group of nonzero elements of Galois field of order 16.

11.2 Proposed Small S-boxes

In this section, we are mainly discussed the algebra of proposed small S-boxes. The following are main steps in constructing proposed S-boxes:

i. We take all invertible elements produced in Eq. (11.1) and element 16 is mapped to 0.

ii. The multiplicative inversion operation in the construction of S-box is the inversion $\mathbb{Z}_{17}^*$ in with the extension 16 ↦ 0. We define the following function $I(x)$ in $\mathbb{Z}_{17}^*$ corresponding to this multiplicative inversion step:

$$I(x) = \begin{cases} x^{-1}, & if\ x < 16 \\ 0, & if\ x = 16. \end{cases} \tag{11.2}$$

We decompose the affine transformation step in proposed S-box construction into two linear transformations L_i $(i = 1, 2)$, two affine transformations K_i $(i = 1, 2)$ and one inversion function function $I(x)$ given as follows:

$$S(x) = K_2 \circ L_2 \circ I \circ K_1 \circ L_1, \tag{11.3}$$

where matrices used in linear and affine transformations are given below:

$$L_1 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix},\ L_2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}, \tag{11.4}$$

$$K_1 = \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix}^T, K_2 = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}^T. \tag{11.5}$$

Now applying permutations of S_4 on Eq. (11.3), we have

$$S - box = S_4(S(x)) = S_4(K_2 \circ L_2 \circ I \circ K_1 \circ L_1). \tag{11.6}$$

There are total 72 S-boxes in total due to three distinct S-boxes were obtained from $\mathbb{Z}_{17}^{*}$ by using Eq. (11.3) and then we apply S_4 permutations on each S-box. We take only three from 72 S-boxes for our projected applications. The proposed S-boxes are presented in Tables (11.1) – (11.3).

Table 11.1: The proposed S-box-I.

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

Table 11.2: The proposed S-box-II.

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

Table 11.3: The proposed S-box-III.

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

11.3 Proposed Algorithm for Steganography Based on Small S-boxes

In this section, we will discussed three different cases of information hiding technique namely steganography based on our three proposed S-boxes.

11.3.1 Steganography Based on S-box-I

We take two color images for secret media and cover media. The color secret image is converted to binary value where each pixel has 8-bit value. We divide our S-box-I into four small blocks with four distinct values and used these small blocks to allocate encoded values for pixels in secret image using corresponding blocks. The small blocks of S-box-I are given as follows:

Table 11.4: Division of S-box-I into four blocks of size 2 × 2.

$$\begin{array}{cc|c} 0 & 1 & \\ \hline 7 & 0 & 0 \\ 1 & 6 & 1 \end{array}$$

Block-I.

$$\begin{array}{cc|c} 0 & 1 & \\ \hline 3 & 13 & 0 \\ 2 & 8 & 1 \end{array}$$

Block-II.

$$\begin{array}{cc|c} 0 & 1 & \\ \hline 14 & 15 & 0 \\ 12 & 4 & 1 \end{array}$$

Block-III.

$$\begin{array}{cc|c} 0 & 1 & \\ \hline 10 & 5 & 0 \\ 9 & 11 & 1 \end{array}$$

Block-IV.

Secondly, we need to select the color component where to embed the secret than take a pixel with 8-bits value which is further distributed into four blocks of two bits (see Fig. 11.1). Each of these two bits block take values from respective blocks (Blocks-I,II,III, IV) in the order of initial part from Block-I, second part from Block-II, third part from Block-III and fourth part from Block-IV respectively (see Fig. 11.2):

$$\underbrace{10}_{b_0} \quad \underbrace{01}_{b_1} \quad \underbrace{01}_{b_2} \quad \underbrace{10}_{b_3}$$

Fig. 11.1: Bit division of secret image pixel.

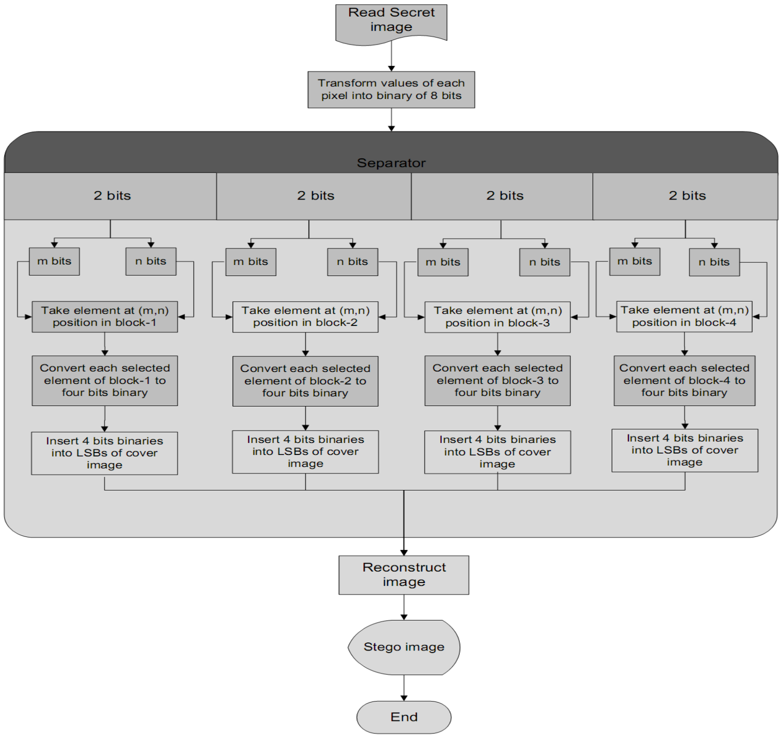

Read Secret image
Transform values of each pixel into binary of 8 bits
Separator
2 bits
2 bits
2 bits
2 bits
m bits
n bits
m bits
n bits
m bits
n bits
m bits
n bits
Take element at (m,n) position in block-1
Take element at (m,n) position in block-2
Take element at (m,n) position in block-3
Take element at (m,n) position in block-4
Convert each selected element of block-1 to four bits binary
Convert each selected element of block-2 to four bits binary
Convert each selected element of block-3 to four bits binary
Convert each selected element of block-4 to four bits binary
Insert 4 bits binaries into LSBs of cover image
Insert 4 bits binaries into LSBs of cover image
Insert 4 bits binaries into LSBs of cover image
Insert 4 bits binaries into LSBs of cover image
Reconstruct image
Stego image
End

Fig. 11. 2: Flow chart for insertion of secret image into cover image.

Embedding Bit into Cover Image

In this phase, we have inserted values obtained from each blocks by bit division into cover image. We converted values of blocks into four bits binaries and placed these four binaries in LSBs of cover image consecutively. First we take the pixels one by one from the cover media and then place 4 bits binaries of LSBs obtained from respective blocks into cover image (see Fig. 11.3-Fig. 11.5).

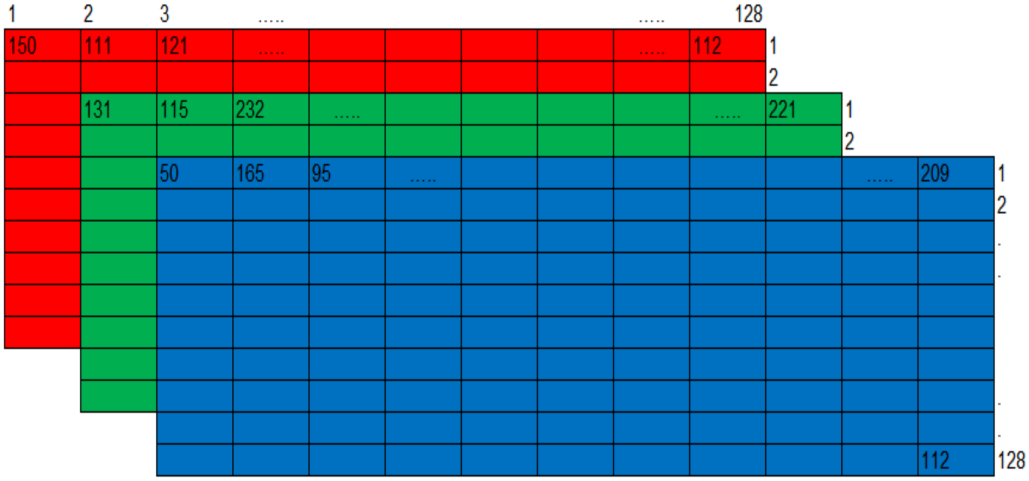


Fig. 11.3: Secret image in 3-D view of size 128 × 128.

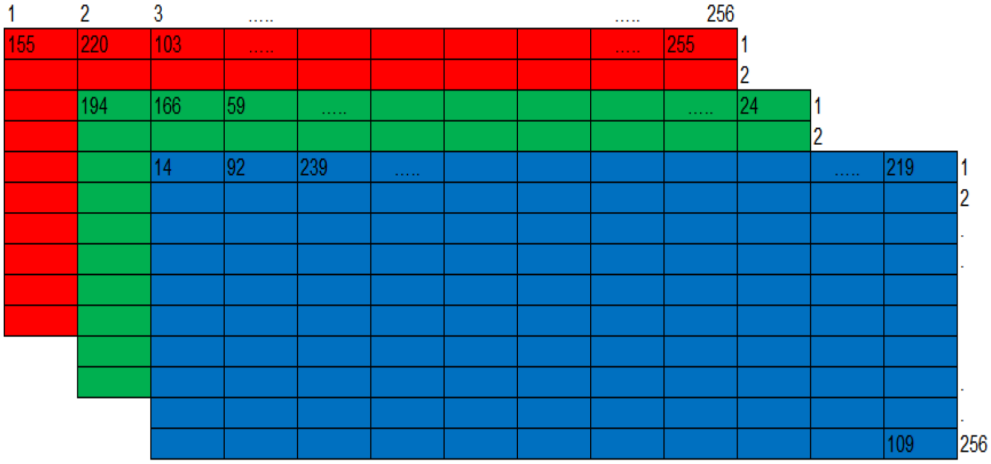


Fig. 11.4: Cover image in 3-D view of size 256 × 256.

$$\begin{array}{ccc}
\underbrace{1001\,\overbrace{1011}}_{155} & \overset{\text{Replace by 1 binaries}}{\Longrightarrow} & \underbrace{1001\,\overbrace{0001}}_{145} \\
\underbrace{1101\,\overbrace{1100}}_{220} & \overset{\text{Replace by 3 binaries}}{\Longrightarrow} & \underbrace{1101\,\overbrace{0011}}_{211} \\
\underbrace{0110\,\overbrace{0111}}_{103} & \overset{\text{Replace by 15 binaries}}{\Longrightarrow} & \underbrace{0110\,\overbrace{1111}}_{111} \\
\underbrace{1111\,\overbrace{1111}}_{255} & \overset{\text{Replace by 9 binaries}}{\Longrightarrow} & \underbrace{1111\,\overbrace{1001}}_{249}
\end{array}$$

Fig. 11.5: Bit insertion into cover image (for red layered).

After getting the new pixel values, we form the stego image. The pixel values for red component 145, 211, 111, 249 are place into the position of the previous values. Similarly we performed these operations for green and blue component of cover image. The resultant stego image in components form is given in Fig. 11.6.

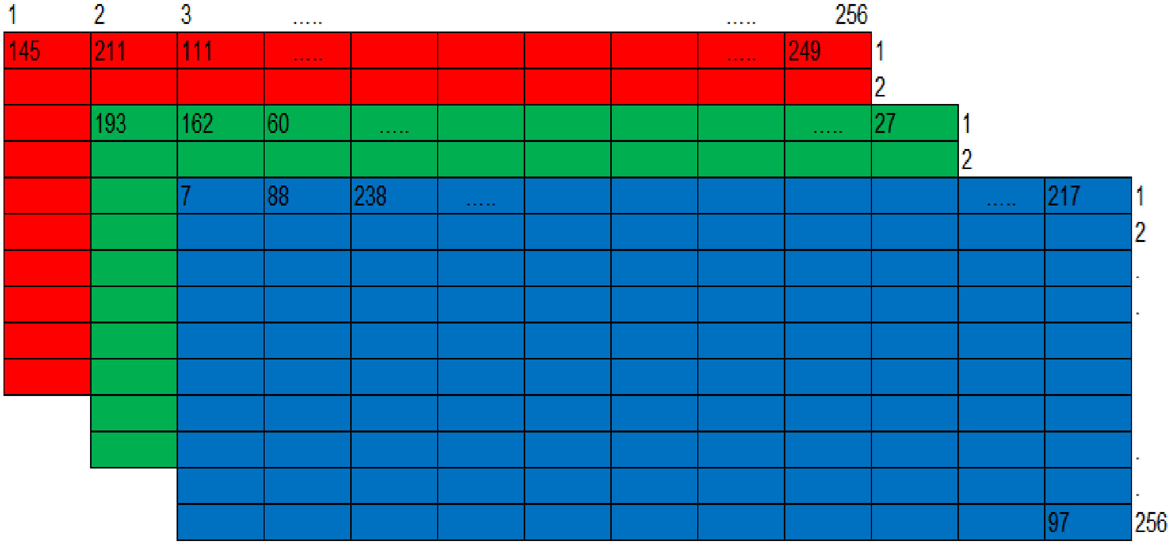
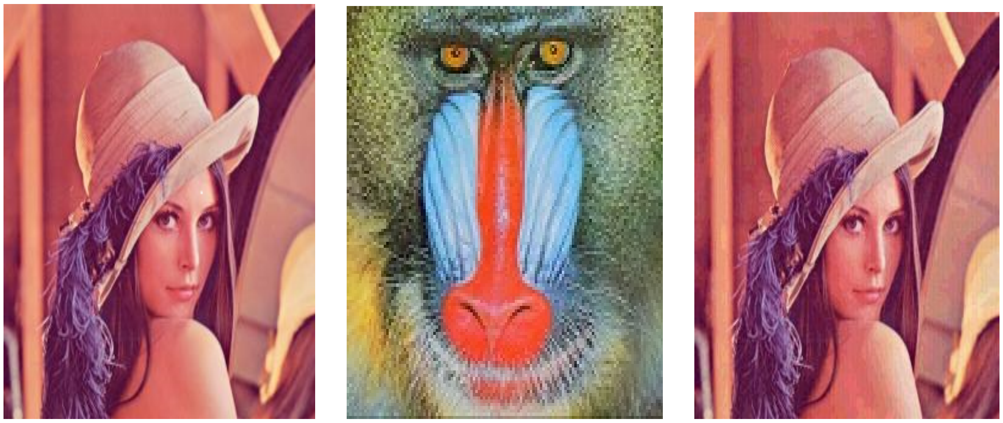


Fig. 11. 6: Stego image in 3-D view of size 256 × 256.

The stego image contents the secret image but we cannot identify the secret image. The changes of the pixel values will be varied from 0 to 15 which is a negligible amount of pixel value due to information carries means LSBs of pixels. The pixels or colors will not be change in large amount with these proposed insertions (see Fig. 11.7). Notice, the difference between the stego-image is barely distinguishable by the human eye.



(a) (b) (c)

Fig. 11.7: (a) Cover (Lena image) of size 256 × 256, (b) Baboon (Secret image) of size 128 × 128, (c) Stego (Lena image) of size 256 × 256.

11.3.2 Steganography Based on S-box-II

In this case, we divide our anticipated S-box-II into two horizontal blocks and each block consists of eight different values belonging from S-box-II (see Table 11.5) and used these small blocks to allocate encoded values for pixels in secret image using corresponding blocks. In next stage, we have performed our experimentation on steganographic media which consists of secret and cover media. The color secret is converted to binary value where each pixel has 8-bits values.

Table 11. 5: Division of S-box-II into two equal parts.

00	01	10	11	
14	0	15	6	0
5	11	10	2	1

Block-I.

00	01	10	11	
9	12	8	4	0
3	1	7	13	1

Block-II.

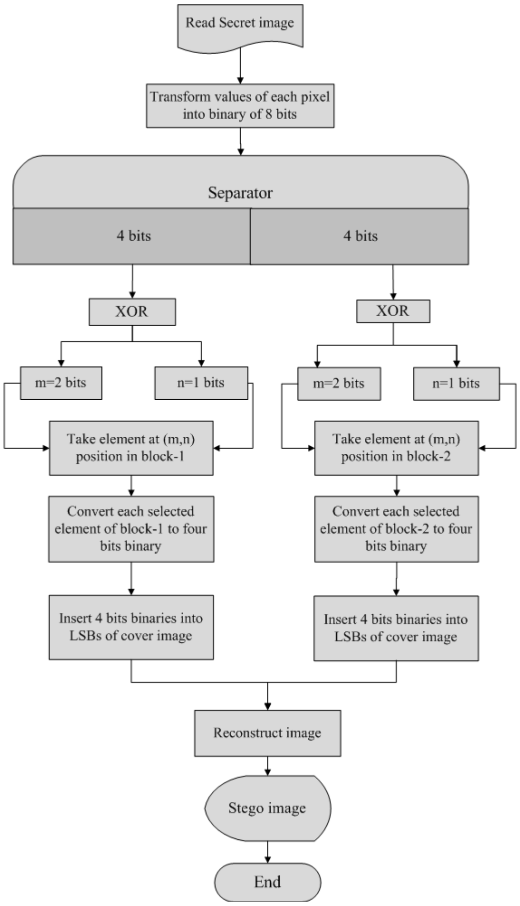
Also, we have to choose the shading part where to insert the mystery than bring a pixel with 8-bits esteem which is further dispersed into two pieces of four bits (see Fig. 11.8). Each of these four bits piece is XOR to three bits so as to take values from segment (Blocks-I,II) in the sequence of starting part from Block-I, second part from Block-II individually (see Fig. 11.9):

$$\underbrace{1001}_{b_0} \qquad \underbrace{0110}_{b_1}$$

$$\underbrace{101}_{c_0} \qquad \underbrace{101}_{c_1}$$

Fig. 11.8: Bit division of secret image pixel into two four bits blocks and XOR operations to each 4 bits blocks.

For instance, we have 101 and 101 binary bits after applying XOR operations on each 4 bits. In first three bits binaries i-e., 101, first two bits 10 represents column and 1 represent row of the blocks. For example 101 represent 2 and in second block we mapped 101 to 13 respectively.



Read Secret image
Transform values of each pixel into binary of 8 bits
Separator
4 bits
4 bits
XOR
XOR
m=2 bits
n=1 bits
m=2 bits
n=1 bits
Take element at (m,n) position in block-1
Take element at (m,n) position in block-2
Convert each selected element of block-1 to four bits binary
Convert each selected element of block-2 to four bits binary
Insert 4 bits binaries into LSBs of cover image
Insert 4 bits binaries into LSBs of cover image
Reconstruct image
Stego image
End

Fig. 11.9: Flow chart for insertion of secret image into cover image.

Embedding Bit into Cover Image

In this phase, we have inserted values obtained from each blocks by bit division into cover image. We convert values of blocks into four bits binaries and placed these four binaries in LSBs of cover image consecutively. First we take the pixels one by one from the cover media and then place 4 LSBs from each of the two horizontal blocks serially (see Figs. 11.11-11.12).

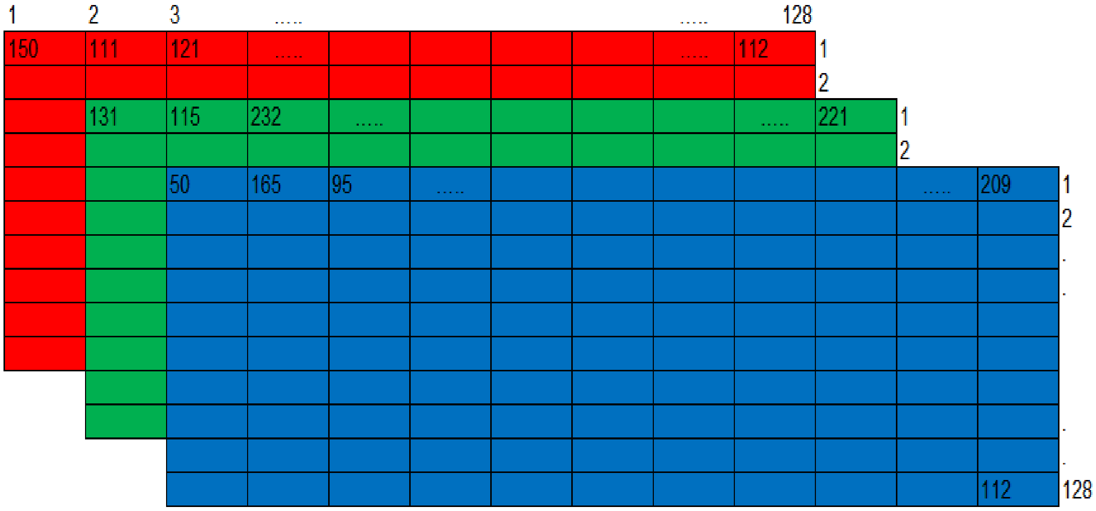


Fig. 11.10: Secret image in 3-D view of size 128 × 128.

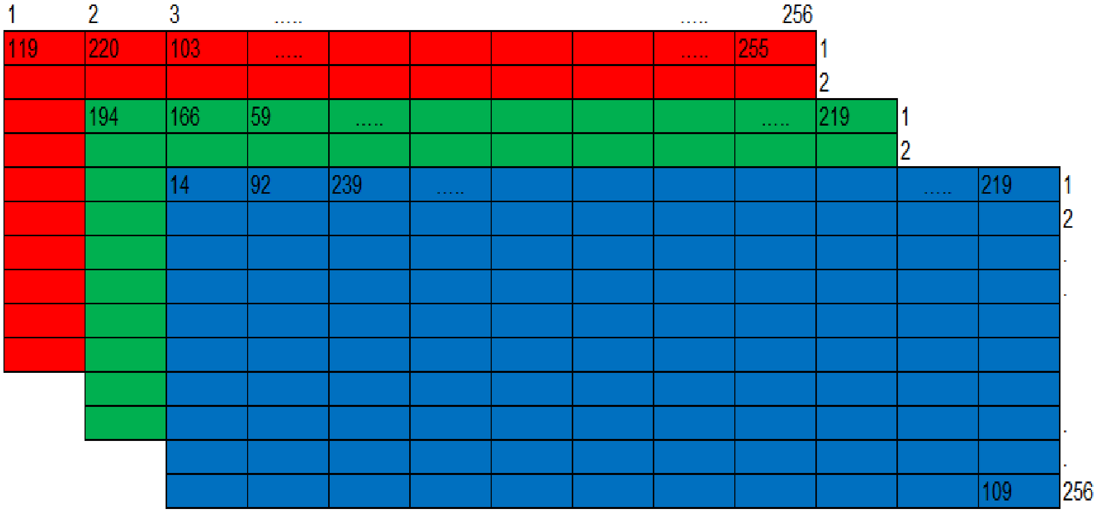


Fig. 11.11: Cover image in 3-D view of size 256 × 256.

$$\boxed{\begin{array}{ccc}
\underbrace{1100\,\overbrace{0011}}_{194} & \underset{\Longrightarrow}{\text{Replace by 5 binaries}} & \underbrace{1100\,\overbrace{0101}}_{197} \\
\underbrace{1101\,\overbrace{1100}}_{166} & \underset{\Longrightarrow}{\text{Replace by 8 binaries}} & \underbrace{1010\,\overbrace{1000}}_{168} \\
\underbrace{0110\,\overbrace{0111}}_{59} & \underset{\Longrightarrow}{\text{Replace by 5 binaries}} & \underbrace{0001\,\overbrace{0101}}_{21} \\
\underbrace{1111\,\overbrace{1111}}_{219} & \underset{\Longrightarrow}{\text{Replace by 8 binaries}} & \underbrace{1101\,\overbrace{1000}}_{104}
\end{array}}$$

Fig. 11.12: Bit insertion into cover image (for green layer).

After getting the new pixel values, we form the stego image. The pixel values for green component 197, 168, 21 and 104 are place into the position of the previous values. Similarly we performed these operations for green and blue component of cover image. The resultant stego image is given in Fig. 11.13.

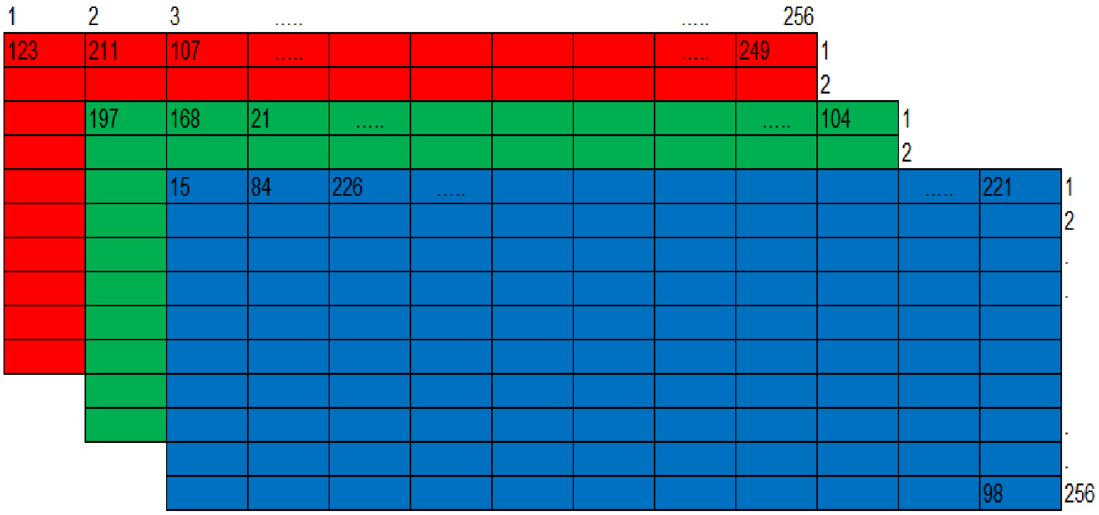
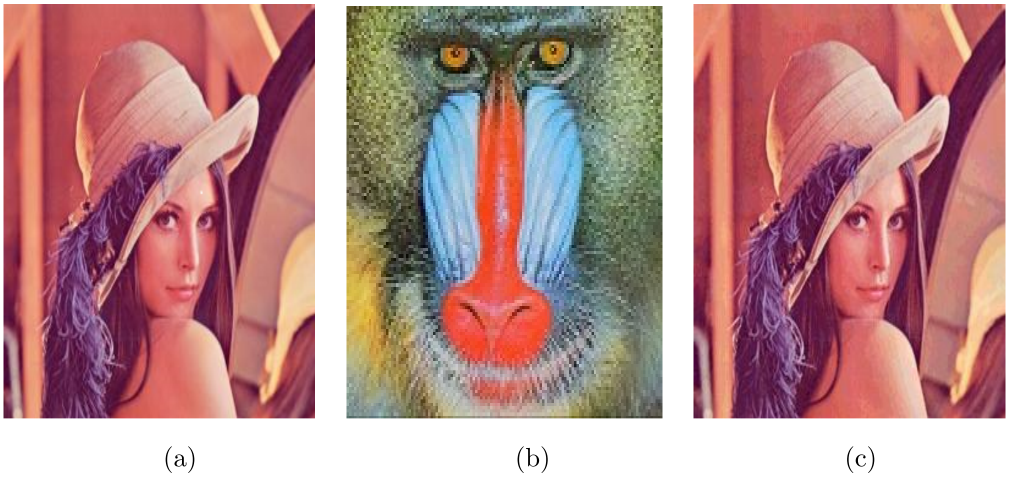


Fig. 11.13: Stego image in 3-D view of size 256 × 256.

The stego image hides the secret image yet we can't distinguish the secret image. The progressions of the pixel qualities will be fluctuated in four bits binaries which is an insignificant portion of pixel esteem because of data conveys implies LSBs of pixels (see Fig. 11.14).



(a) (b) (c)

Fig. 11.14: (a) Cover (Lena image) of size 256 × 256, (b) Baboon (Secret image) of size 128 × 128, (c) Stego (Lena image) size 256 × 256.

11.3.3 Steganography Based on S-box-III

The method projected on S-box-III is fundamentally based on division of S-box-III into two vertical small blocks with eight distinct values and used these small blocks to allocate encoded values for pixels in secret image using corresponding blocks. The vertical blocks of S-box-III are given as follows:

Table 11.6: Division of S-box-III into two equal parts.

0	1		0	1	
13	0	00	12	15	00
8	5	01	4	6	01
3	11	10	2	9	10
7	1	11	14	10	11
Block-I			Block-II		

Secondly, we need to select the color component where to embed the secret than take a pixel with 8-bits value which is further distributed into two blocks of four bits (see Fig. 11.15). Each of these four bits block is XOR to three bits in order to take values from respective blocks (Blocks-I,II) in the order of initial part from Block-I, second part from Block-II respectively (see Fig. 11.15):

$$\underbrace{1001}_{b_0} \qquad \underbrace{0110}_{b_1}$$

$$\underbrace{101}_{c_0} \qquad \underbrace{101}_{c_1}$$

Fig. 11.15: Bit division of secret image pixel into two four bits blocks and XOR operations to each 4 bits block.

For instance, we have 101 and 101 binary bits after applying XOR operations on each 4 bits. In first three bits binaries i-e., 101, first two bits 10 represents row and 1 represent column of the blocks. For example 101 represent 5 and in second block we mapped 101 to 9 respectively.

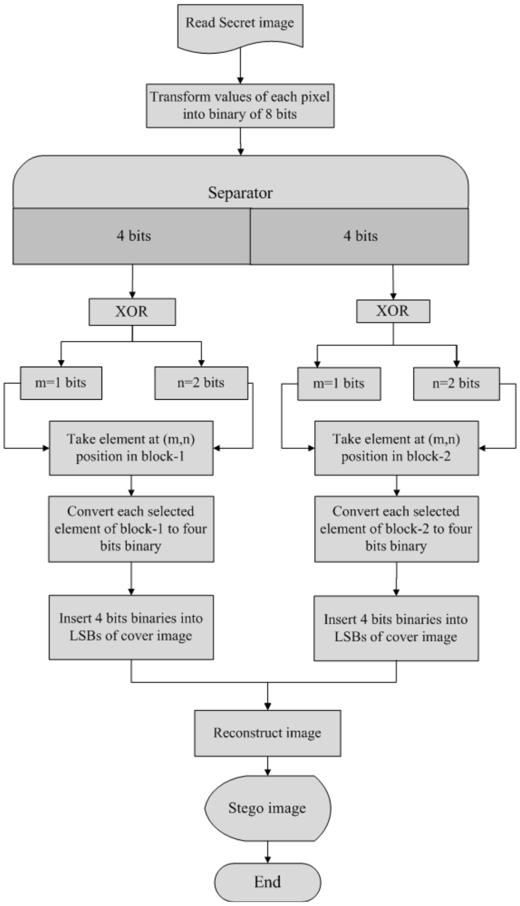

Read Secret image
Transform values of each pixel into binary of 8 bits
Separator
4 bits
4 bits
XOR
XOR
m=1 bits
n=2 bits
m=1 bits
n=2 bits
Take element at (m,n) position in block-1
Take element at (m,n) position in block-2
Convert each selected element of block-1 to four bits binary
Convert each selected element of block-2 to four bits binary
Insert 4 bits binaries into LSBs of cover image
Insert 4 bits binaries into LSBs of cover image
Reconstruct image
Stego image
End

Fig. 11.16: Flow chart for insertion of secret image into cover image.

Embedding Bit into Cover image

In this stage, we have embedded qualities gotten from every blocks by bit division into cover image as in previous two cases. We transformed decimal values of blocks into four bits pairs and set these four bits in LSBs of cover image successively. In the first place, we take the pixels one by one from the cover media and after that place 4 LSBs from each of the two vertical blocks serially (see Figs. 11.11-11.12).

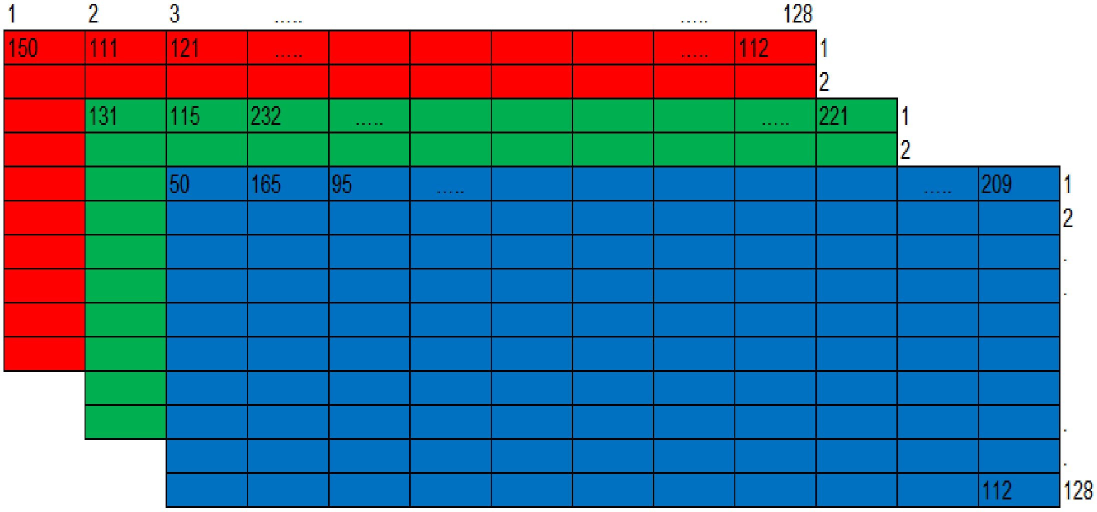


Fig. 11.17: Secret image in 3-D view of size 128 × 128.

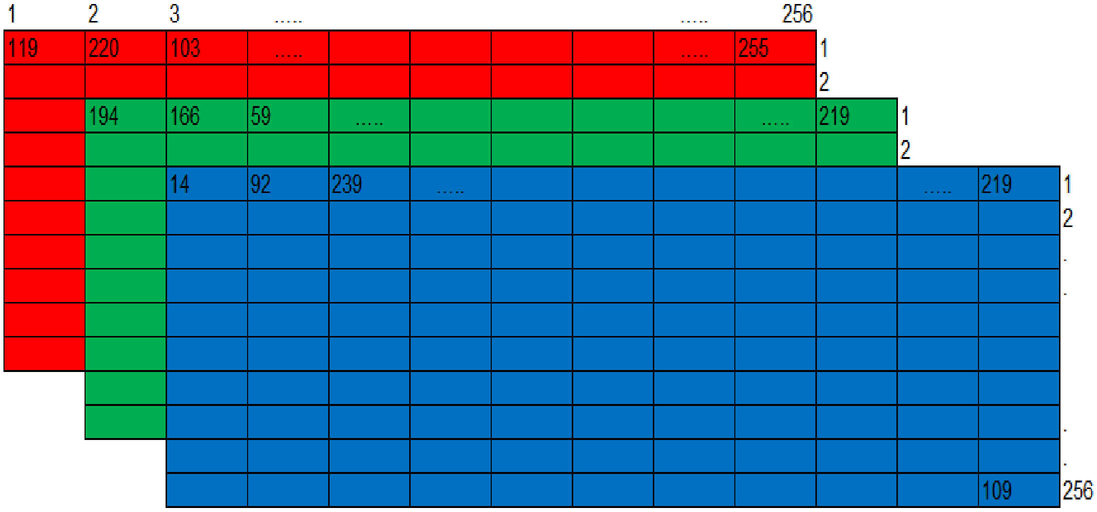


Fig. 11.18: Cover image in 3-D view of size 256 × 256.

$$\underbrace{0000\,\overbrace{1110}}_{14} \quad \overset{\text{Replace by 3 binaries}}{\Longrightarrow} \quad \underbrace{0000\,\overbrace{0011}}_{3}$$

$$\underbrace{1101\,\overbrace{1100}}_{92} \quad \overset{\text{Replace by 14 binaries}}{\Longrightarrow} \quad \underbrace{0101\,\overbrace{1110}}_{94}$$

$$\underbrace{1110\,\overbrace{1111}}_{239} \quad \overset{\text{Replace by 1 binaries}}{\Longrightarrow} \quad \underbrace{1110\,\overbrace{0001}}_{225}$$

$$\underbrace{1101\,\overbrace{1011}}_{219} \quad \overset{\text{Replace by 10 binaries}}{\Longrightarrow} \quad \underbrace{1101\,\overbrace{1010}}_{218}$$

$$\underbrace{0110\,\overbrace{1101}}_{109} \quad \overset{\text{Replace by 1 binaries}}{\Longrightarrow} \quad \underbrace{0110\,\overbrace{0001}}_{97}$$

Fig. 11.19: Bit insertion into cover image (for blue layer).

After getting the new pixel values, we form the stego image. The pixel values for red component 145, 211, 111, 249 are place into the position of the previous values. Similarly we performed these operations for green and blue component of cover image. The resultant stego image is given in Figs. 11.20 − 11.21.

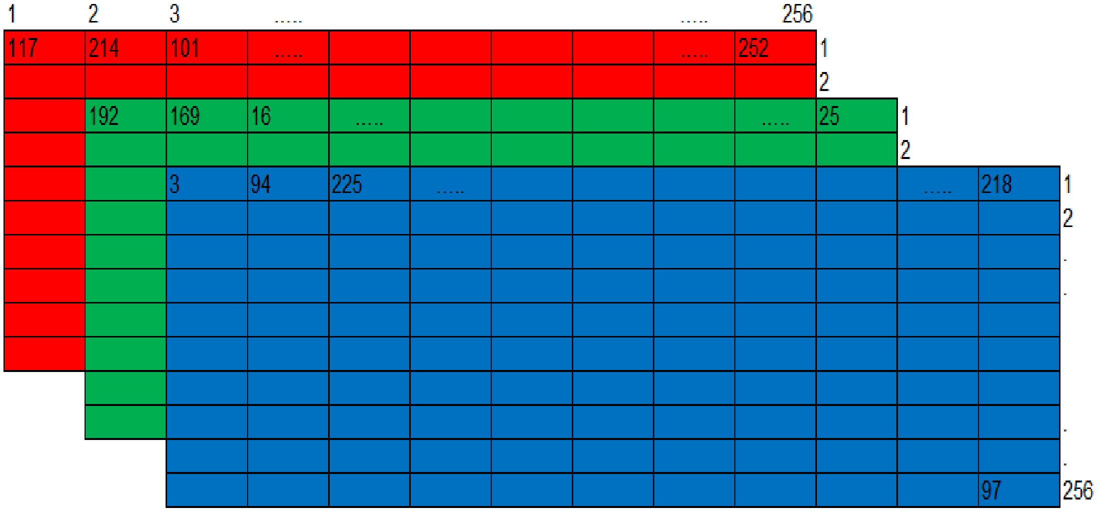
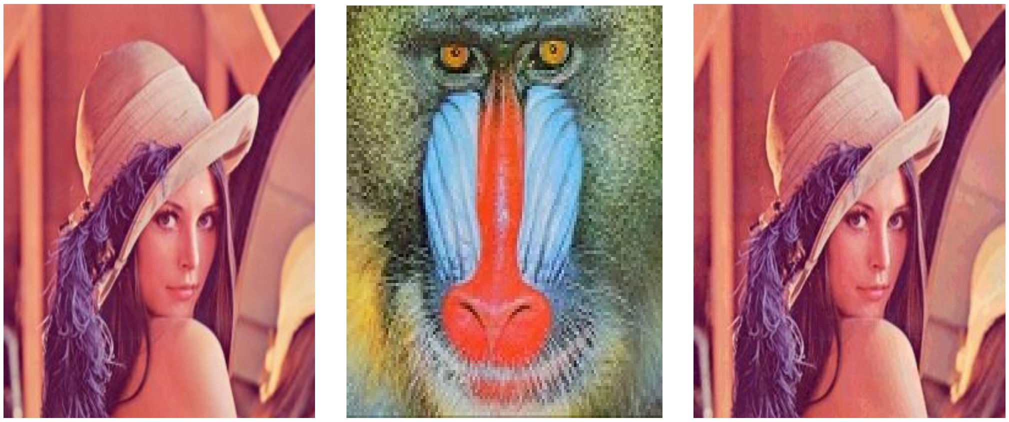


Fig. 11.20: Stego image in 3-D view of size 256 × 256.

The stego image contents the secret image but we cannot identify the secret image. The changes of the pixel values will be varied from 0 to 15 which is a negligible amount of pixel value due to information carries means LSBs of pixels. So the pixels or colors will not be change in large amount.



(a) (b) (c)

Fig. 11.21: (a) Cover (Lena image) of size 256 × 256, (b) Baboon (Secret image) of size 128 × 128, (c) Stego (Lena image) of size 256 × 256.

11.4 Statistical Analyses

11.4.1 First Order Texture Features

First-order texture measures are determined from the original image values. They do not consider the connections with neighborhood pixel. Histogram-based approach to composition analysis is in light of the intensity esteem focuses on all or part of an image spoke to as a histogram. Characteristics got from this approach incorporate moments such as mean, standard deviation, skewness and kurtosis [182]. The histogram of intensity levels will be a straightforward outline of the measurable data of the image and individual pixels will be utilized to compute the gray-level histogram. Along these lines, the histogram contains the first-order measurable data about the picture (or sub picture). These measurements are defined as follows:

$$\begin{aligned} Mean &= \mu_k = \frac{\sum_{i=0}^{M-1}\sum_{i=0}^{N-1} I_k(i,j)}{M \times N}, && (11.7)\\ Standard\ Deviation &= \sigma_k = \sqrt{\frac{\sum_{i=0}^{M-1}\sum_{i=0}^{N-1} \left(I_k(i,j)-\mu\right)^2}{M \times N}}, && (11.8)\\ Skewness &= \gamma_1 = \frac{\sum_{i=0}^{M-1}\sum_{i=0}^{N-1} \left(I_k(i,j)-\mu\right)^3}{M \times N \times \sigma^2}, && (11.9)\\ Kurtosis &= \gamma_2 = \frac{\sum_{i=0}^{M-1}\sum_{i=0}^{N-1} \left(I_k(i,j)-\mu\right)^4}{M \times N \times \sigma^4} - 3. && (11.10) \end{aligned}$$

The proposed system will be likewise assessed built with respect to first order texture features like mean, standard deviation, skewness and kurtosis to authenticate the effect on image in case of replacement of bits [181]. Here, we lead an investigation between cover image and the stego-image in light of statistical alteration. The aftereffect of the execution parameters prior and then afterward the implanting procedure are ascertained and precise in (see Tables 11.7-11.9). The image parameters are estimation of the safety for the stego-system. Minimizing parameters distinction is one of the fundamental targets in order to get rid of statistical attacks.

Table 11.7: First order texture analysis for steganographic system based on S-box-I.

	Cover image			Stego image		
	Red	Green	Blue	Red	Green	Blue
Mean	0.574219	0.304688	0.171875	0.578125	0.300781	0.175781
Standard Deviation	0.495429	0.461177	0.378011	0.494826	0.459496	0.38138
Skewness	-0.300201	0.848678	1.73946	-0.316386	0.868817	1.70357
Kurtosis	1.09012	1.72025	4.02573	1.1001	1.75484	3.90216

Table 11.8: First order texture analysis for steganographic system based on S-box-II.

	Cover image			Stego image		
	Red	Green	Blue	Red	Green	Blue
Mean	0.566406	0.300781	0.175781	0.566406	0.304688	0.199219
Standard Deviation	0.496541	0.459496	0.38138	0.496541	0.461177	0.400195
Skewness	-0.267999	0.868817	1.70357	-0.267999	0.848678	1.50612
Kurtosis	1.07182	1.75484	3.90216	1.07182	1.72025	3.26839

Table 11.9: First order texture analysis for steganographic system based on S-box-III.

	Cover image			Stego image		
	Red	Green	Blue	Red	Green	Blue
Mean	0.589844	0.3164406	0.179688	0.59375	0.304688	0.1875
Standard Deviation	0.492825	0.465984	0.384679	0.492094	0.461177	0.391077
Skewness	-0.365321	0.789526	1.66861	-0.381771	0.848678	1.66128
Kurtosis	1.13346	1.62335	3.78427	1.14575	1.72025	3.5641

From the Tables 11.7-11.9, it is seen that there is no significant difference between the mean, standard deviation, skewness and kurtosis between the cover-image and the stego-image. This study shows that the magnitude of change in stego-image based on image parameters is small from a cover image. These analysis confirmed the reliability of proposed information hiding scheme.

11.4.2 Second Order Texture Features

The second order features are based on gray level co-occurrence matrix (GLCM) [143] and it is one of the most popular methods for pixel variation statistics. Some of the second order statistical features are entropy, contrast, homogeneity, energy and correlation of the gray level pixels, defined as [147].

Entropy

This statistic measures the disorder or complexity of an image. The entropy is large when the image is not texturally uniform and many GLCM elements have very small values. Complex textures tend to have high entropy. Entropy is strongly, but inversely correlated to energy.

Angular Second Moment

This statistic is also called uniformity or angular second moment or energy. It measures the textural uniformity that is pixel pair repetitions. It detects disorders in textures. Energy reaches a maximum value equal to one. High energy values occur when the gray level distribution has a constant or periodic form. Energy has a normalized range. The GLCM of less homogeneous image will have large number of small entries.

Inertia

This statistic measures the spatial frequency of an image and is difference moment of GLCM. This measure is also called contrast. It is the difference between the highest and the lowest values of a contiguous set of pixels. It measures the amount of local variations present in the image. A low contrast image presents GLCM concentration term around the principal diagonal and features low spatial frequencies.

Inverse Difference Moment

Inverse difference moment is the local homogeneity. It is high when local gray level is uniform and inverse GLCM is high. Inverse difference moment weight value is the inverse of the Contrast weight. It measures image homogeneity as it assumes larger values for smaller gray tone differences in pair elements. It is more sensitive to the presence of near diagonal elements in the GLCM. It has maximum value when all elements in the image are same. GLCM contrast and homogeneity are strongly, but inversely, correlated in terms of equivalent distribution in the pixel pairs population. It means homogeneity decreases if contrast increases while energy is kept constant.

Correlation

Correlation is a measure of gray tone linear dependencies in the image, in particular, the direction under investigation is the same as vector displacement. High correlation values imply a linear relationship between the gray levels of pixel pairs. Thus, GLCM correlation is uncorrelated with GLCM energy and entropy, i.e., to pixel pairs repetitions. Correlation reaches it maximum regardless of pixel pair occurrence, as high correlation can be measured either in low or in high energy situations. The five common textures features are given as follows:

$$\mathit{Entropy} = -\sum_{i=0}^{N_g-1}\sum_{i=0}^{N_g-1} P_{i,j}\log P_{i,j}, \tag{11.11}$$

$$\mathit{Angular\ second\ moment} = \sum_{i=0}^{N_g-1}\sum_{i=0}^{N_g-1} P_{i,j}^2, \tag{11.12}$$

$$\mathit{Inertia} = \sum_{i=0}^{N_g-1}\sum_{i=0}^{N_g-1}(i-j)^2 P_{i,j}, \tag{11.13}$$

$$\mathit{Inverse\ difference\ moment} = \sum_{i=0}^{N_g-1}\sum_{i=0}^{N_g-1}\frac{P_{i,j}}{1+(i-j)^2}, \tag{11.14}$$

$$\mathit{Correlation} = \frac{\sum_{i=0}^{N_g-1}\sum_{i=0}^{N_g-1} ijP_{i,j}-\mu_x\mu_y}{\sigma_x\sigma_y}, \tag{11.15}$$

where $P_{i,j}$ is the (i,j) th entry of the normalized co-occurrence matrix, N_g is the number of gray levels of an image, μ_x, μ_y, σ_x and σ_y are the means and standard deviations of the marginal probabilities $P_x(i)$ and $P_y(j)$ obtained by summing up the rows or the columns of matrix $P_{i,j}$ respectively. A complete second order texture analyses of proposed steganographic techniques are presented in Tables 11.10−11.12.

Table 11.10: Second order texture analysis for steganographic system -I.

	Cover image			Stego image		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.357274	0.374341	0.341988	0.363067	0.377068	0.359574
Homogeneity	0.87546	0.873758	0.879416	0.871784	0.872185	0.870501
Entropy	7.27854	7.57291	7.05544	7.28795	7.4984	7.07227
Correlation	0.926586	0.932406	0.861158	0.924661	0.931912	0.855571
Energy	0.141417	0.100946	0.17250	0.139518	0.100174	0.163683

Table 11.11: Second order texture analysis for steganographic system -II.

	Cover image			Stego image		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.372687	0.392816	0.365273	0.384053	0.399004	0.3799381
Homogeneity	0.872453	0.871262	0.874949	0.868679	0.869308	0.869657
Entropy	7.2911	7.58133	7.07945	7.29576	7.5216	7.08361
Correlation	0.923453	0.929416	0.853858	0.920758	0.928464	0.848844
Energy	0.138624	0.0999494	0.169877	0.138927	0.0990417	0.165287

Table 11.12: Second order texture analysis for steganographic system -III.

	Cover image			Stego image		
	Red	Green	Blue	Red	Green	Blue
Contrast	0.776379	0.752543	0.761933	0.822595	0.79784	0.807721
Homogeneity	0.827889	0.836525	0.834285	0.816972	0.827534	0.822203
Entropy	7.76992	7.86391	7.72703	7.77383	7.84679	7.74232
Correlation	0.92185	0.90929	0.888301	0.91677	0.903984	0.882108
Energy	0.0892355	0.0758187	0.0851768	0.085572	0.073779	0.0804689

11.4.3 Image Quality Measures

Image quality measures are key for most picture handling applications. Any picture and feature procurement framework can utilize the quality metric to modify itself consequently for getting enhanced quality pictures. It can be utilized to pose as a viable rival and assess picture handling systems and algorithms.

Image quality measures are prevailing to convey quantitative information on the dependability of extracted images. Commonly the nature of an image combination system will be assessed utilizing numerical procedures which endeavor to measure loyalty utilizing image to image examinations, a few image quality measurements have been created to foresee the unmistakable contrasts between cover and stego images. This work is taking into account the way that concealing data in computerized media obliges changes of the sign properties that present some type of debasement, regardless of how little; these degradations can go about as marks that could be utilized to uncover the presence of a hidden message (see Tables 11.13 – 11.21). Image quality measurements are sorted into six groups as per the kind of data they are utilizing. The classifications utilized are:

i. Pixel Difference-based measures,

ii. Correlation-based measures,

iii. Edge-based measures,

iv. Spectral Distance-based measures,

v. Context-based measures,

vi. Human Visual System based measures.

The pixel difference-based measures were derived based on pixel to pixel error such as mean square error(MSE), root mean square error (RMSE), average difference (AD), maximum difference (MD), mean absolute error (MAE), peak signal to noise ratio(PSNR), signal to noise ratio, enhancement error (EME) and mutual information (MI). The correlation based measures includes normalized cross correlation (NCC), structure content (SC) and universal image quality index (UIQI), structural similarity index metric (SSIM) are included in human visual system-based measures. The pixel difference-based, correlation-based and human visual system-based measures defined as follows :

Mean Squared-Error (MSE)

The mean-squared-error (MSE) is the simplest, and the most widely used, full -reference image quality measurement. Similarity is determined by computing the error between the stego image and the reference cover image.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j) - S(i,j))^2, \tag{11.16}$$

where $M \times N$ is the size of the image. The parameters $C(i,j)$ and $S(i,j)$ refer to the pixels located at the ith row and the jth column of original image and stego image due to the imbedding of the secret information. The mean square error (MSE) represents the cumulative squared error between the stego image and cover image. A lower figure of MSE conveys lower error/distortion between the cover and stego image.

Root Mean Square Error (RMSE)

To evaluate the proposed stegosystem, this method is tested on the color Lena image of 256×256 pixels. To find the accuracy of the results and the robustness of the steganographic system, a root mean square of error is calculated. These criteria provide the error between cover image and stego image. The rms value can be described by the following relation:

$$RMSE = \sqrt{\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j) - S(i,j))^2}, \tag{11.17}$$

where the $C(i,j)$ is the pixel intensity of the cover image, $S(i,j)$ is the pixel intensity of the stego image. The row and column numbers of these two images are defined by $M \times N$.

Mean Absolute Error (MAE)

MAE is average of absolute difference between the reference signal and test image. It is given by the equation

$$MAE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| C(i,j) - S(i,j)) \right|. \tag{11.18}$$

Average Difference (AD)

AD is simply the average of difference between the reference signal and the test image and it is given by the equation

$$AD = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j) - S(i,j)). \tag{11.19}$$

Maximum Difference (MD)

MD is the maximum of the error signal (difference between the stego and cover image)

$$MD = Max \left| C(i,j) - S(i,j)) \right|. \tag{11.20}$$

Peak Signal to Noise Ratio (PSNR)

The PSNR is evaluated in decibels and is inversely proportional the Mean Squared Error. It is given by the equation

$$PSNR = 10 \log_{10} \frac{255^2}{\sqrt{MSE}}. \tag{11.21}$$

Enhancement Error (EME)

A number of blind reference metrics have been proposed during the last decade. EME (enhancement error) has been developed by [183] give an absolute score to each image on the basis of image contrast processed with Fechner's Law relating contrast to perceived brightness or the well-known entropy concept. The following equation give us the EME formula

$$EME = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} 20 \log_2 \frac{\max(I(i,j)}{\min(I(i,j)}, \tag{11.22}$$

where the image is divided into $M \times N$ blocks, $\max(I(i,j), \min(I(i,j)$ are the maximum and minimum values of the pixels in each block of the enhanced image.

Structure Content (SC)

It is one of the correlation based measures. It means the closeness between two digital images which can be quantified in terms of correlation function. This metrics measures the similarity between two images.

The structural content metric is based on the following equations

$$SC = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j))^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (S(i,j))^2}, \tag{11.23}$$

where $C(i,j)$ is the $(i,j)^{th}$ pixel value of cover image, $S(i,j)$ is the pixel value of stego image.

Normalized Cross-Correlation (NCC)

The normalized cross-correlation (NCC) metric is the metric that is used to show the amount of deflection in the stego image with respect to the cover image after insertion of the message. The normalized cross-correlation (NCC) is applied to evaluate the performance of various existing methods which is given by the following equation

$$NCC = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} C(i,j) \times S(i,j)}{\sum_{i=1}^{M} \sum_{j=1}^{N} (C(i,j))^2}. \tag{11.24}$$

This measure measures the similarity between two images, hence in this sense it's complementary to the difference-based measures.

Human Visual Systems (HVS) Based Measures

A major emphasis in recent research has been given to a deeper analysis of the human visual system (HVS) features. Researchers assume that incorporating knowledge of the human visual system (HVS) and human perception into objective quality assessment algorithms could increase their accuracy. This HVS-based framework has been the dominant paradigm for the last three decades. The underlying premise is that humans do not perceive images as signals in a high-dimensional space, but are interested in various attributes of those images, such as brightness, contrast, shape and texture of objects, orientations, smoothness, etc. Since the sensitivity of the HVS is different for different aspects of images, it makes sense to account for these sensitivities while making a comparison between the test and the reference signal.

There are a lot of HVS characteristics that may influence the human visual perception on image quality. Although HVS is too complex to fully understand with present psychophysical means, the incorporation of even a simplified model into objective measures reportedly leads to a better correlation with the response of the human observers. Human visual system (HVS) has been extensively exposed to the natural visual environment, and a variety of evidence has shown that the HVS is highly adapted to extract useful information from natural scenes. Two human visual systems (HVS) based image quality measures are given below:

Universal Image Quality Index (UIQI) Let $x = \{x_i, i = 1, 2, \ldots\ldots.., N\}$ and $y = \{y_i, i = 1, 2, \ldots\ldots\ldots.., N\}$ be the cover and stego images. The proposed quality index is defined as:

$$Q = \frac{4\sigma_{xy}\overline{xy}}{(\sigma_x^2 + \sigma_y^2)(\overline{x}^2 + \overline{y}^2)}, \tag{11.25}$$

where

$$\begin{aligned} \overline{x} &= \frac{1}{N}\sum_{i=1}^{N} x_i, \;\; \overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \;\; \sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2, & (11.26) \\ \sigma_y^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \overline{y})^2, \;\; \sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}). & (11.27) \end{aligned}$$

The dynamic range of Q is $[0,1]$. The best value $Q = 1$, is achieved when $x_i = y_i, \; i = 1, 2, ..., n$. The universal image quality index can also be defined as the product of three components:

$$Q = Q_1 \times Q_2 \times Q_3, \tag{11.28}$$

where

$$Q_1 = \frac{\sigma_{xy}}{\sigma_x \sigma_y}, \tag{11.29}$$

$$Q_2 = \frac{2\overline{xy}}{\overline{x}^2 + \overline{y}^2}, \tag{11.30}$$

$$Q_3 = \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2}, \tag{11.31}$$

where first term defines the degree of correlation between x and y with dynamic range between $[-1,1]$, second term measures how close the luminance is between x and y range is $[0,1]$ and third term measures how similar the contrasts of the image x and y are.

Structural Similarity Index Metric (SSIM) The structural similarity (SSIM) index is a technique for measuring the similarity between two pictures. The SSIM record is a full reference metric; as such, the measuring of picture quality focused around an introductory uncompressed or without distortion picture as reference. SSIM is intended to enhance conventional strategies like peak signal to noise ratio (PSNR) and mean squared error (MSE), which have turned out to be conflicting with human eye recognition. The contrast as for different strategies specified at one time, for example, MSE or PSNR is that these methodologies appraisal perceived errors; then again, SSIM considers image corruption as perceived change in structural data. Structural data is the way to go that the pixels have solid between conditions particularly when they are spatially close. These conditions convey essential data about the structure of the objects in the visual scene. The SSIM metric is figured on different windows of a picture. The measure between original and marked images of size is [165]:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{11.32}$$

where μ_x is the average of X, μ_y is the average of Y, σ_x^2 is the variance of , σ_y^2 is the variance of and σ_{xy} is covariance of X and Y, $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables which is to stabilize the weak denominator, L is the dynamics range of the pixels-values.

Table 11.13: Simple statistics errors for cover and stego images for steganopgraphic system-I.

	Image color components		
	Red	Green	Blue
MSE	14.6721	13.0356	13.6659
RMSE	3.83042	3.61049	3.69674
PSNR	36.4659	36.9795	36.7744
MAE	3.00061	2.84267	2.87642
AD	0.554657	0.561356	0.409073
MD	19	15	20
NAE	0.0166466	0.028655	0.0272485
EME (Cover image)	7.87436	19.4497	11.7173
EME (Stego image)	8.34606	20.9805	12.7009

Table 11.14: Simple statistics errors for cover and stego images for steganographic system-II.

	Image color components		
	Red	Green	Blue
MSE	25.8806	21.7254	27.3693
RMSE	5.0873	4.66105	5.23156
PSNR	34.0011	34.7611	33.7552
MAE	3.93118	3.57509	4.0278
AD	-0.537994	-0.58934	-0.755493
MD	28	28	29
NAE	0.0218192	0.0361022	0.0382225
EME (Cover image)	8.15026	18.8942	12.4686
EME (Stego image)	8.47044	19.8622	12.9848

Table 11.15: Simple statistics errors for cover and stego images for steganographic system-III.

	Image color components		
	Red	Green	Blue
MSE	26.6135	25.9582	27.5116
RMSE	6.29393	6.07933	6.44295
PSNR	32.1524	32.4537	31.9491
MAE	4.67821	4.49437	4.81732
AD	-0.664719	-0.69764	-1.10873
MD	40	36	37
NAE	0.0303843	0.0403222	0.0515037
EME (Cover image)	16.1939	14.9457	18.2867
EME (Stego image)	17.5367	16.6970	19.2368

The error comparison between cover image and stego image are shown in the tables 11.13-11.15. These results indicate the presence of secret information in stego image. These tables additionally concludes that the stego image is of better quality if MAE, MSE, RMSE, AD, NAE and EME values are less while the high value of PSNR means that the stego image is most similar to original image. It is hard for the human eyes to distinguish between cover image and stego image when the PSNR ratio is larger than 30dB. The values of MD for image color components simply indicates the presence of hidden data in stego image pixels. The values of MD equals to zero for no secret information in stego image or in other words stego image is not generated from cover image by apply the proposed steganographic technique.

Table 11.16: Correlation based image quality measures for steganographic system-I.

	Image color components		
	Red	Green	Blue
SC	1.00593	1.00926	1.00561
NCC	0.996841	0.994891	0.996658

Table 11.17: Correlation based image quality measures for steganographic system-II.

	Image color components		
	Red	Green	Blue
SC	0.994433	0.990126	0.93674
NCC	1.00243	1.00412	1.00561

Table 11.18: Correlation based image quality measures for steganographic system-III.

	Image color components		
	Red	Green	Blue
SC	0.993695	0.918738	0.912193
NCC	1.00249	1.00322	1.00718

The value of normalized cross correlation is equal to one for same images i-e., the outcome of proposed technique on cover image with itself and the hidden data length equals to zero. That means the minimum value of normalized cross correlation value equals to zero, in other words when the normalized cross correlation value equals to (zero) that means there is no hidden data in the image. When the value of normalized cross correlation is greater than one that means the two images (cover image, stego image) are not identical , in other words the stego image is carrying hidden data (secret message), and the hidden data length is greater than (zero) (see tables 11.16 − 11.18). As it can be seen from Figs. 11.7, 11.14 and 11.21, while the two images seem similar to each other, in fact they are different in the structure. The analyses of cover and stego images with respect to structure content reveals that two images under study seem to be same for human eyes but in fact they are not, and these values represents the similarity factor between cover and stego images (see tables 11.16 − 11.18).

Table 11.19: Human visual system based image quality measures for steganographic system-I.

	Image color components		
	Red	Green	Blue
UIQI	0.833114	0.848125	0.841500
SSIM	0.937995	0.944472	0.937726

Table 11.20: Human visual system based image quality measures for steganographic system-II.

	Image color components		
	Red	Green	Blue
UIQI	0.80084	0.833829	0.78798
SSIM	0.919597	0.930689	0.911057

Table 11.21: Human visual system based image quality measures for steganographic system-III.

	Image color components		
	Red	Green	Blue
UIQI	0.846898	0.863393	0.859568
SSIM	0.912866	0.918783	0.912193

The universal image quality index split the judgment of similarity between cover image (C) and stego image (S) into three comparisons: Luminance, Contrast and Structural Information. SSIM estimates "Perceived change in structural information". It computes the similarity between two images of common size. The value of UIQI and SSIM varies between 1 and -1. Closer the highest positive value denotes too much less change in two images and -1 shows totally mismatch. The UIQI and SSIM are considered as more consistent and accurate than MSE and PSNR. As MSE and PSNR are adequate for image closeness measure just when the image vary by essentially expanding contortion of a certain sort. In any case they neglect to catch image quality when they are utilized to gauge across contortion sorts. SSIM is broadly utilized technique for estimation of image quality. It meets expectations precisely can quantify better across distortion types when differentiated with MSE and PSNR. The numerical values of UIQI and SSIM are close to one which clearly indicates that the suggested technique is highly secure for transferring secret information in information carrier and two images seem similar to each other but actually they are different in the structure.

11.5 Conclusion

We have utilized the small S-boxes that take into account Galois field $\mathbb{Z}_{17}^{*}$, symmetry group S_4 and LSB to get a protected stage-image. The proposed system has been utilized for applications that oblige high-volume inserting with strength against certain statistical attacks. The present system is an endeavor to recognize the prerequisites of a decent information concealing algorithm. Our results display that the LSB insertion utilizing small S-boxes are superior to straightforward LSB insertion. The image resolution doesn't change much and is unimportant when we embed the message into the image and the image is secured. In this manner, it is unrealistic to harm the data by unapproved personnel. This paper concentrates on the methodology like increasing the security of the secret embedded information and reducing the distortion rate.

Chapter 12

A Novel Cryptosystem Based on General Linear Group

We have developed a novel public-key cryptosystem that uses large abelian subgroup of general linear group over residue ring. The merit of this proposed public key cryptosystem is that we can select session key in abelian subgroup of general linear group which reduces exponentiations and performed encryption effectively in a simple way. Our algorithm doesn't use matrix modular exponentiation which leads us to problem in implementing. The aim here is to decrease the number of these exponentiations and consequently to accelerate the execution of encryption algorithm. A discussion about the security of built modifications made in the article shows that the level of security is high enough for an appropriate choice of parameters of the cryptosystems.

12.1 Subgroup of General Linear Group

Let H be the subgroup of the of general linear group of degree 2 i.e., $GL(2,\mathbb{Z}_n)$ which is given as follows:

$$H=\left\{\begin{pmatrix} a_1 & b_1 \\ b_1 & a_1 \end{pmatrix}\middle|\ a_1,b_1\in\mathbb{Z}_n \text{ and } a_1^2-b_1^2\neq 0\right\}, \tag{12.1}$$

which shows that elements of subgroup H are belong to unit group of residue ring $\mathbb{Z}_n$ that is $\mathbb{Z}_n^*$. The subgroup H is an abelian subgroup of the group $GL(2,\mathbb{Z}_n)$ which be verified easily:

1. Let $H_1=\begin{pmatrix} a_1 & b_1 \\ b_1 & a_1 \end{pmatrix}$, $H_2=\begin{pmatrix} c_1 & d_1 \\ d_1 & c_1 \end{pmatrix}\in H$, we have to show that $H_1H_2=H_2H_1$. Now we have to multiply both matrices, i.e.,

   $$M=H_1H_2=\begin{pmatrix} a_1c_1+b_1d_1 & a_1d_1+b_1c_1 \\ a_1d_1+b_1c_1 & a_1c_1+b_1d_1 \end{pmatrix},\ \text{ and } \det(H_1H_2)=\det(H_1)\det(H_2)\in\mathbb{Z}_n^*.$$

2. Let $H_1 = \begin{pmatrix} a_1 & b_1 \\ b_1 & a_1 \end{pmatrix} \in H,\ \det(H_1) = \mu \in \mathbf{Z}_n^*$, we have $H_1^{-1} \ \in H$, because

$$H_1^{-1} = \begin{pmatrix} \mu^{-1}a & \mu^{-1}b \\ \mu^{-1}b & \mu^{-1}a \end{pmatrix},\ \det(H_1^{-1}) = \mu^{-1} \in \mathbf{Z}_n^*,$$

3. Let $H_1 = \begin{pmatrix} a_1 & b_1 \\ b_1 & a_1 \end{pmatrix}, H_2 = \begin{pmatrix} c_1 & d_1 \\ d_1 & c_1 \end{pmatrix} \in H$, we have $H_1H_2 = H_2H_1$,

$$H_1H_2 = \begin{pmatrix} a_1c_1 + b_1d_1 & a_1d_1 + b_1c_1 \\ a_1d_1 + b_1c_1 & a_1c_1 + b_1d_1 \end{pmatrix},$$

$$= \begin{pmatrix} c_1d_1 + d_1b_1 & c_1b_1 + d_1a_1 \\ c_1b_1 + d_1a_1 & c_1a_1 + d_1b_1 \end{pmatrix},$$

$$= H_2H_1.$$

Let a and b be the random elements of the ring $\mathbb{Z}_n$. Let M be the corresponding element in the ring $M_2(\mathbb{Z}_n)$. What is the probability of the case that M is not in the group H? We will answer this question with respect to two cases of n. Let us consider each case separately:

Case I

$$n = rs, \text{where } r \text{ and } s \text{ are different primes.} \tag{12.2}$$

The cardinality of the residue ring and its unit group is given as follows:

$$|\mathbb{Z}_n| = rs,\ |\mathbb{Z}_n^*| = \varphi(n) = (r-1)(s-1),\ \text{where } \varphi(n) \text{ is an Euler function.} \tag{12.3}$$

Then the probability P the case that matrix M is not in the group H is given below:

$$Prob = 1 - \frac{\varphi(n)}{n} = 1 - \frac{(r-1)(s-1)}{rs} = \frac{1}{r} + \frac{1}{s} - \frac{1}{rs}. \tag{12.4}$$

If bit length of primes p and q will be greater than or equal to 90 then we have

$$Prob \leq 2^{-89}. \tag{12.5}$$

Case II

$$n = r^l, \text{where } r \text{ is primes and } l \geq 2. \tag{12.6}$$

The cardinality of the residue ring and its unit group is given as follows:

$$|\mathbb{Z}_n| = r^l,\ |\mathbb{Z}_n^*| = \varphi(n) = r^{l-1}(r-1),\ \text{where } \varphi(n) \text{ is an Euler function.} \tag{12.7}$$

Then the probability P in the case that matric M is not in the group H is given below:

$$Prob = 1 - \frac{\varphi(n)}{n} = 1 - \frac{r^{l-1}(r-1)}{r^l} = \frac{1}{r}. \tag{12.8}$$

If bit length of primes p and q will be greater than or equal to 90 then we have

$$Prob \leq 2^{-90}. \tag{12.9}$$

In both cases probability is negligible small and therefore one may suppose that in both cases the random matrix M over $\mathbb{Z}_n$ with overwhelming probability is in the subgroup H.

12.2 Proposed Cryptosystem Based on Units of Residue Ring and General Linear Group of Degree 2

In this section, we are mainly discussed two cryptosystems which are based on two cases defined in previous section. We will discussed two cryptosystem in detailed about their key generation, encryption and decryption algorithm.

12.2.1 Cryptosystem-I

Key Generation

User A, doing the following:

1. Select random prime numbers r and s such that $r \neq s$ and computes $n = rs \; or \; n = r^l$ for $l \geq 2$.

2. Select four random integers a, b, c and $d \in \mathbb{Z}_n$.

3. Make two matrices from four integers selected in step 2;

$$A = \begin{pmatrix} a & b \\ b & a \end{pmatrix}, \quad B = \begin{pmatrix} c & d \\ d & c \end{pmatrix}.$$

4. Verify the membership of randomly selected in group H. If at least one of the matrix is not belongs to H then return to step 2.

5. User A defines two commutative inner product automorphisms of the ring $M_2(\mathbb{Z}_n)$:

$$\chi : V \to A^{-1}VA, \qquad \delta : V \to B^{-1}VB, \tag{12.10}$$

for every matric $A \in M_2(\mathbb{Z}_n)$.Since matrices A and $B \in H$ and therefore automorphism χ and δ commute.

6. User A computes the following automorphisms of the ring $M_2(\mathbb{Z}_n)$:

$$\rho \ = \ \chi^2\delta, \qquad \sigma = \chi\delta^2, \tag{12.11}$$

$$\rho \ : \ V \to (A^2B)^{-1}V(A^2B), \qquad \sigma : V \to (AB^2)^{-1}V(AB^2). \tag{12.12}$$

   Automorphisms ρ and σ commute and $\rho = \chi\delta^{-1}\sigma, \ \sigma = \chi^{-1}\delta\rho$.

7. User A select a random invertible matrix $N \in GL(2, \mathbb{Z}_n)$, such that N does not belong to group H.

8. Computes the matrices:

$$N^{-1} \ , \rho(N), \quad \sigma(N^{-1}) \ . \tag{12.13}$$

9. User A public key is

$$\left(n, \rho(N), \sigma(N^{-1})\right), \tag{12.14}$$

   and private key

$$(A, B). \tag{12.15}$$

Encryption

User B will performed the following tasks:

1. Represents the plaintext m as a sequence of 2×2 matrices over residue ring $\mathbb{Z}_n$:

$$m^{(1)}, m^{(2)}, m^{(3)}, ..., m^{(k)}.$$

2. For every $m^{(i)}$ $(i = 1, 2, ..., k)$, choose a random matrix $X^{(i)} \in H$.

3. Define for every $i = 1, 2, ..., k$, the automorphisms

$$\vartheta^{(i)} : V \to (X^{(i)})^{-1}V(X^{(i)}), \tag{12.16}$$

   for every $V \in M_2(\mathbb{Z}_n)$.

4. Computes for every $i = 1, 2, ..., k$ matrices $\vartheta^{(i)}(\rho(N))$, $\vartheta^{(i)}(\sigma(N^{-1}))$ and $m^{(i)}\vartheta^{(i)}(\rho(N))$.

5. Selects for every $i = 1, 2, ..., k$ random unit $\mu \in \mathbb{Z}_n$ and computes the ciphertext:

$$\begin{aligned} C &= \left(C^{(1)}, C^{(2)}, ..., C^{(k)}\right), \qquad C^{(i)} = (C_1^{(i)}, C_2^{(i)}), && (12.17)\\ C_1^{(i)} &= \mu_i^{-1}\vartheta^{(i)}(\sigma(N^{-1})), \qquad C_2^{(i)} = \mu_i m^{(i)}\vartheta^{(i)}(\rho(N)),\ i = 1, 2, ..., k. && (12.18)\end{aligned}$$

Decryption

User A will follow the steps given below for deciphering enciphered message:

1. Computes for every $i = 1, 2, ..., k$ using the private key:

$$d^{(i)} = \chi^{-1}\delta(C_1^{(i)}) = \chi^{-1}\delta(\mu_i^{-1}\vartheta^{(i)}(\sigma(N^{-1}))). \tag{12.19}$$

2. Computes for every $i = 1, 2, ..., k$ matrices:

$$m^{(i)} = C_2^{(i)} d^{(i)} = (\mu_i m^{(i)} \vartheta^{(i)}(\rho(N)))d^{(i)}. \tag{12.20}$$

3. Finally, User A can easily recovers the original message i.e., plaintext m from the matrix sequences $m^{(1)}, m^{(2)}, m^{(3)}, ..., m^{(i)}$.

12.2.2 Cryptosystem-II

Key Generation

User A, doing the following:

1. Select a random prime numbers p and q such that $p \neq q$ and computes $n = pq$ or $n = p^r$ for $r \geq 2$.

2. Select random matrix $D \in GL(2, \mathbb{Z}_n)$.

3. Computes the following matrices

$$I = D^2,\ J = D^3,\ I^2J \text{ and } IJ^2. \tag{12.21}$$

4. Select a random matrix $N \in GL(2, \mathbb{Z}_n)$.

5. Define the automorphism

$$\begin{aligned} \rho &= \chi^2\delta, \qquad \sigma = \chi\delta^2, && (12.22)\\ \rho &: V \rightarrow (I^2J)^{-1}V(I^2J), \qquad \sigma : V \rightarrow (IJ^2)^{-1}V(IJ^2). && (12.23)\end{aligned}$$

for every matric $V \in M_2(\mathbb{Z}_n)$.The automorphisms χ, δ, ρ and σ commute with each other because the corresponding matrices I, J, I^2J and IJ^2 are some integral exponents of matrix D.

6. Computes the matrices:
$$IJ\ , \rho(N), \quad \sigma(N^{-1})\ . \tag{12.24}$$

7. User A public key is
$$\big(n, IJ, \rho(N), \sigma(N^{-1})\big)\ , \tag{12.25}$$
and private key
$$(I, J). \tag{12.26}$$

Encryption

User B will performed the following tasks:

1. Represents the plaintext m as a sequence of 2×2 matrices over residue ring $\mathbb{Z}_n$:
$$m^{(1)}, m^{(2)}, m^{(3)}, ..., m^{(k)}. \tag{12.27}$$

2. For every $m^{(i)}$ $(i = 1, 2, ..., k)$, select a random integer k_i computes matrix
$$X^{(i)} = (IJ)^{k_i}. \tag{12.28}$$

3. Define for every $i = 1, 2, ..., k$, the automorphisms
$$\vartheta^{(i)} : V \to (X^{(i)})^{-1}V(X^{(i)}), \tag{12.29}$$
for every $V \in M_2(\mathbb{Z}_n)$.

4. Computes for every $i = 1, 2, ..., k$ matrices $\vartheta^{(i)}(\rho(N))$, $\vartheta^{(i)}(\sigma(N^{-1}))$ and $m^{(i)}\vartheta^{(i)}(\rho(N))$.

5. Selects for every $i = 1, 2, ..., k$ random units $\mu \in \mathbb{Z}_n^*$ and computes the ciphertext:
$$\begin{aligned} C &= \Big(C^{(1)}, C^{(2)}, ..., C^{(k)}\Big)\ , & C^{(i)} &= (C_1^{(i)}, C_2^{(i)}), & (12.30)\\ C_1^{(i)} &= \mu_i^{-1}\vartheta^{(i)}(\sigma(N^{-1})), & C_2^{(i)} &= \mu_i m^{(i)}\vartheta^{(i)}(\rho(N)),\ i = 1, 2, ..., k. & (12.31)\end{aligned}$$

Decryption

User A will follow the steps given below for deciphering enciphered message:

1. Computes for every $i = 1, 2, ..., k$ using the private key:

$$d^{(i)} = \chi^{-1}\delta(C_1^{(i)}) = \chi^{-1}\delta(\mu_i^{-1}\vartheta^{(i)}(\sigma(N^{-1}))). \tag{12.32}$$

2. Computes for every $i = 1, 2, ..., k$ matrices:

$$m^{(i)} = C_2^{(i)}d^{(i)} = (\mu_i m^{(i)}\vartheta^{(i)}(\rho(N)))d^{(i)}. \tag{12.33}$$

3. Finally, User A can easily recovers the original message i.e., plaintext m from the matrix sequences $m^{(1)}, m^{(2)}, m^{(3)}, ..., m^{(i)}$.

12.3 Implementations of Proposed Cryptosystem-I and Cryptosystem-II

In this section, we have constructed examples to the mentioned cryptosystem-I and cryptosystem-II respectively.

12.3.1 Example

Key Generation

User A will do the following steps:

1. Select two prime numbers say $p = 2$ and $q = 13$ and compute $n = pq = 26$.

2. Select four random integers in the modular ring $\mathbb{Z}_{26}$, i.e., $5, 4, 6, 1$.

3. Make matrices from the integers selected in step 2, i.e.,

$$A = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}, \ B = \begin{pmatrix} 6 & 1 \\ 1 & 6 \end{pmatrix}. \tag{12.34}$$

4. Computes $\det A = 9$, $\det B = 9$ and then computes $(\det A)^{-1} = 3$ and $(\det B)^{-1} = 3$, therefore A and B are units in the matrix ring $M_2(\mathbb{Z}_{26})$.

5. Defines two automorphisms of the ring $M_2(\mathbb{Z}_{26})$:

$$\chi : V \to A^{-1}VA, \qquad \delta : V \to B^{-1}VB, \tag{12.35}$$

for every matrix $V \in M_2(\mathbb{Z}_{26})$.

6. Define two automorphisms of the ring $M_2(\mathbb{Z}_{26})$:

$$\begin{aligned} \rho &= \chi^2\delta, & \sigma = \chi\delta^2, \quad (12.36)\\ \rho &: V \to (AB^2)^{-1}V(AB^2), & \sigma : V \to (A^2B)^{-1}V(A^2B). \quad (12.37) \end{aligned}$$

7. User A select a random invertible matrix $N \in GL(2, \mathbb{Z}_{26})$, such that N does not belong to group H.

$$N = \begin{pmatrix} 7 & 2 \\ 1 & 3 \end{pmatrix}, N^{-1} = \begin{pmatrix} 7 & 4 \\ 15 & 25 \end{pmatrix}. \quad (12.38)$$

8. Computes the matrices:

$$\rho(N) = (AB^2)^{-1}N(AB^2) \quad = \begin{pmatrix} 7 & 2 \\ 1 & 3 \end{pmatrix}, \quad (12.39)$$

$$\sigma(N^{-1}) = (A^2B)^{-1}N^{-1}(A^2B) = \begin{pmatrix} 25 & 15 \\ 4 & 7 \end{pmatrix}. \quad (12.40)$$

9. User A public key is

$$\left(n = 26, \rho(N) = \begin{pmatrix} 7 & 2 \\ 1 & 3 \end{pmatrix}, \sigma(N^{-1}) = \begin{pmatrix} 25 & 15 \\ 4 & 7 \end{pmatrix}\right), \quad (12.41)$$

and private key

$$\left(A = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}, \ B = \begin{pmatrix} 6 & 1 \\ 1 & 6 \end{pmatrix}\right). \quad (12.42)$$

Encryption

User B will performed the following steps:

1. Presents the plaintext as a matrix $m \in M_2(\mathbb{Z}_{26})$:

$$m = \begin{pmatrix} 11 & 3 \\ 9 & 3 \end{pmatrix}, \quad (12.43)$$

2. Select the random matrix $X = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$ and its inverse is $X^{-1} = \begin{pmatrix} 2 & 19 \\ 19 & 2 \end{pmatrix}$.

3. Define automorphism ϑ of the ring $M_2(\mathbb{Z}_{26})$:

$$\vartheta : V \to X^{-1}VX, \tag{12.44}$$

for every $V \in M_2(\mathbb{Z}_{26})$.

4. Compute the matrices:

$$\begin{aligned} \vartheta(\rho(N)) &= X^{-1}\rho(N)X = \begin{pmatrix} 11 & 17 \\ 12 & 25 \end{pmatrix}, & (12.45) \\ \vartheta(\sigma(N^{-1})) &= X^{-1}\sigma(N^{-1})X = \begin{pmatrix} 17 & 24 \\ 21 & 15 \end{pmatrix}, & (12.46) \end{aligned}$$

5. Now select a unit element of $\mathbb{Z}_{26}$ randomly:

$$\mu = 9,\ \mu^{-1} = 3. \tag{12.47}$$

6. Computes the ciphertext

$$\begin{aligned} C &= (C_1, C_2). & (12.48) \\ C_1 &= \mu^{-1}\vartheta(\sigma(N^{-1})) = \begin{pmatrix} 25 & 20 \\ 11 & 19 \end{pmatrix}, & (12.49) \\ C_2 &= \mu m\vartheta(\rho(N)) = \begin{pmatrix} 9 & 18 \\ 19 & 24 \end{pmatrix}. & (12.50) \end{aligned}$$

Decryption

User A will performed the following steps in order to recover the plaintext:

1. Computes the matrix d, using the private key:

$$d = \chi^{-1}\delta(C_1) = (A^{-1}B)^{-1}C_1(A^{-1}B) = \begin{pmatrix} 19 & 11 \\ 20 & 25 \end{pmatrix}. \tag{12.51}$$

2. Computes the following matric manipulations to get the final plaintext:

$$m = C_2 d = \begin{pmatrix} 11 & 3 \\ 9 & 3 \end{pmatrix}. \tag{12.52}$$

12.3.2 Example

Key generation

User A will do the following steps:

1. Select a prime number say $p = 7$ and compute $n = p^2 = 49$.

2. Select a random matrix

$$D = \begin{pmatrix} 7 & 3 \\ 5 & 2 \end{pmatrix} \in GL(2, \mathbb{Z}_{49}). \tag{12.53}$$

3. Computes matrices

$$I = D^2 = \begin{pmatrix} 15 & 27 \\ 45 & 19 \end{pmatrix}, \quad J = D^3 = \begin{pmatrix} 14 & 18 \\ 18 & 26 \end{pmatrix}, \quad I^2J = \begin{pmatrix} 14 & 24 \\ 40 & 23 \end{pmatrix}, \quad IJ^2 = \begin{pmatrix} 22 & 41 \\ 3 & 19 \end{pmatrix} \tag{12.54}$$

4. User A select a random invertible matrix $N \in GL(2, \mathbb{Z}_{49})$:

$$N = \begin{pmatrix} 9 & 2 \\ 7 & 3 \end{pmatrix}, N^{-1} = \begin{pmatrix} 4 & 30 \\ 7 & 12 \end{pmatrix}.$$

5. Defines automorphisms of the ring $M_2(\mathbb{Z}_{49})$:

$$\chi : V \to I^{-1}VI, \qquad \delta : V \to J^{-1}VJ, \tag{12.55}$$

   for every matric $V \in M_2(\mathbb{Z}_{49})$. Now computes the automorphisms

$$\rho = \chi^2\delta, \qquad \sigma = \chi\delta^2, \tag{12.56}$$

$$\rho : V \to (I^2J)^{-1}V(I^2J), \qquad \sigma : V \to (IJ^2)^{-1}V(IJ^2). \tag{12.57}$$

6. Calculating the following matrices:

$$IJ = \begin{pmatrix} 19 & 31 \\ 19 & 0 \end{pmatrix}, \quad \rho(N) = \begin{pmatrix} 4 & 5 \\ 43 & 8 \end{pmatrix}, \quad \sigma(N^{-1}) = \begin{pmatrix} 2 & 34 \\ 20 & 14 \end{pmatrix}. \tag{12.58}$$

7. User A public key is

$$\left(n = 49, \ \rho(N) = \begin{pmatrix} 4 & 5 \\ 43 & 8 \end{pmatrix}, \quad \sigma(N^{-1}) = \begin{pmatrix} 2 & 34 \\ 20 & 14 \end{pmatrix}, \quad IJ = \begin{pmatrix} 19 & 31 \\ 19 & 0 \end{pmatrix}\right), \tag{12.59}$$

and private key

$$\left(I = \begin{pmatrix} 15 & 27 \\ 45 & 19 \end{pmatrix}, \quad J = \begin{pmatrix} 44 & 1 \\ 18 & 26 \end{pmatrix}\right). \tag{12.60}$$

Encryption

User B will performed the following steps:

1. Presents the plaintext as a matrix $m \in M_2(\mathbb{Z}_{49})$:

$$m = \begin{pmatrix} 10 & 1 \\ 3 & 1 \end{pmatrix} \in M_2(\mathbb{Z}_{49}), \tag{12.61}$$

2. Select the random integer k for instance $k = 3$, and computes the matrix:

$$X = (IJ)^3 = \begin{pmatrix} 37 & 1 \\ 18 & 19 \end{pmatrix}. \tag{12.62}$$

3. Define automorphism ϑ of the ring $M_2(\mathbb{Z}_{49})$:

$$\vartheta : V \to X^{-1}VX, \tag{12.63}$$

for every $V \in M_2(\mathbb{Z}_{49})$.

4. Compute the matrices:

$$\vartheta(\rho(N)) \quad = \quad X^{-1}\rho(N)X = \begin{pmatrix} 4 & 5 \\ 43 & 8 \end{pmatrix}, \tag{12.64}$$

$$\vartheta(\sigma(N^{-1})) \quad = \quad X^{-1}\sigma(N^{-1})X = \begin{pmatrix} 2 & 34 \\ 20 & 14 \end{pmatrix}. \tag{12.65}$$

5. Now select a unit element of $\mathbb{Z}_{49}$ randomly:

$$\mu = 11, \ \mu^{-1} = 9. \tag{12.66}$$

6. Computes the ciphertext

$$
\begin{aligned}
C &= (C_1, C_2). && (12.67)\\
C_1 &= \mu^{-1}\vartheta(\sigma(N^{-1})) = \begin{pmatrix} 10 & 7 \\ 22 & 36 \end{pmatrix}, && (12.68)\\
C_2 &= \mu m\vartheta(\rho(N)) = \begin{pmatrix} 3 & 22 \\ 31 & 43 \end{pmatrix}. && (12.69)
\end{aligned}
$$

Decryption

User A will performed the following steps in order to recover the plaintext:

1. Computes the matrix d, using the private key:

$$d = \chi\delta^{-1}(C_1) = (IJ^{-1})^{-1}C_1(IJ^{-1}) = \begin{pmatrix} 23 & 44 \\ 4 & 23 \end{pmatrix}.$$

2. Computes the following matric manipulations to get the final plaintext:

$$m = C_2 d = \begin{pmatrix} 10 & 1 \\ 3 & 1 \end{pmatrix}. \qquad (12.70)$$

Theorem 108 *The decryption in Cryptosystem-1 and Cryptosystem-2 are correct.*

Proof. *Automorphisms $\vartheta^{(i)}, i = 1, ..., k$ commute with the automorphisms χ and δ in both cryptosystems. The automorphisms defined in both cryptosystems are different. Let us start with the following computations:*

$$
\begin{aligned}
(\mu_i m^{(i)}\vartheta^{(i)}(\rho(N)))d^{(i)} &= ((\mu_i m^{(i)}\vartheta^{(i)}(\rho(N))))(\chi\delta^{-1}(\mu_i^{-1}\vartheta^{(i)}(\sigma(N^{-1})))),\\
&= (\mu_i\mu_i^{-1}m^{(i)}\vartheta^{(i)}(\rho(N))))(\vartheta^{(i)}(\chi\delta^{-1}(\sigma(N^{-1})))),\\
&= m^{(i)}(\vartheta^{(i)}(\rho(N)))(\vartheta^{(i)}((\rho(N^{-1}))),\\
&= m^{(i)}\vartheta^{(i)}(\rho(N)\rho(N^{-1})),\\
&= m^{(i)}\vartheta^{(i)}(\rho(I)),\\
&= m^{(i)}I,\\
&= m^{(i)}. && (12.71)
\end{aligned}
$$

■

12.4 Some attacks on proposed cryptosystems

12.4.1 A ciphertext only attack

Let $C = (C_1, C_2)$ be a ciphertext for the plaintext $m^{(i)}$, then

$$C_1 = \mu_i^{-1}\vartheta^{(i)}(\sigma(N^{-1})), \quad C_2 = \mu m\vartheta(\rho(N)), \tag{12.72}$$

and therefore we come to the equation system with unknowns matrices m, X and unknown unit element $\mu \in \mathbb{Z}^*$:

$$\begin{cases} C_1 = \mu_i^{-1}X^{-1}(\sigma(N^{-1}))X, \\ C_2 = \mu mX^{-1}(\rho(N))X. \end{cases} \tag{12.73}$$

For random unit μ cryptanalyst has not another way to solve this equation system as to suppose concrete value $\mu_0 = \mu$ and to solve the conjugation problem: to find unknown matrix X from the equation:

$$\mu_0 C_1 = X^{-1}(\sigma(N^{-1}))X. \tag{12.74}$$

Rewriting this matrix equation as system of four linear equations with four unknowns cryptanalyst finds the set of solutions, depending on one or more parameters, each of which runs $\mathbb{Z}_n$ .Then he inserts each solution $X = X_0$ in the second equation of system :

$$C_2 = \mu_0 mX_0^{-1}(\rho(N))X_0. \tag{12.75}$$

and finds corresponding solution $m = m_0$. Thus, for each fixed μ_0 cryptanalyst receives at least n pairs of the form (X_0, m_0). Because μ_0 accepts $\phi(n)$ values, the cryptanalyst gets $n\phi(n)$ triples of the form (μ_0, X_0, m_0). Consequently, if n is not less than 64 bit integer, then check which of these non less approximately 2^{125} triplets is a true solution becomes infeasible.

12.4.2 A known-plaintext attack

Let $((m^{(i)}, C^{(i)}), i = 1, 2, 3, ..., k)$ be the pairs of the form plaintext-ciphertext. Cryptanalyst needs to find unknown plaintext $m^{(k+1)}$ from the corresponding ciphertext $C^{(k+1)}$. In our case for the cryptosystems 1 and 2 encryption uses the new random one-time key X for the new plaintext. Therefore knowledge of previous pairs of the form plaintext-ciphertext gives no information to find the unknown plaintext from the corresponding ciphertext for a new pair.

12.4.3 A chosen-ciphertext attack

Let m^* be a random matrix in the group $GL(2, \mathbb{Z}_n)$ and $C = (C_1, C_2)$ be a ciphertext for unknown plaintext m. Cryptanalyst Connor computes m^*C and offers User A to decrypt the ciphertext $C^* = (C_1, m^*C_2)$. Then User A finds corresponding plaintext m^*m and sends it to Connor. Finally Connor computes the initial plaintext as the following:

$$(m^*)^{-1}(m^*m) = m. \tag{12.76}$$

12.4.4 Protection from a chosen ciphertext attack

To prevent this attack one has to replace one-sided ciphertext with two-sided ciphertext. Namely, one-sided ciphertext:

$$C = (C_1, C_2), \ \ C_1 = \mu^{-1}X^{-1}(\sigma(N^{-1}))X, \ C_2 = \ \mu m X^{-1}(\rho(N))X, \tag{12.77}$$

is replaced with two-sided ciphertext

$$C = (C_1, C_2), \ \ C_1 = \mu^{-1}X^{-1}(\sigma(N^{-1}))X, \ C_2 = \ \mu^2(X^{-1}(\rho(N))X)mX^{-1}(\rho(N))X. \tag{12.78}$$

In this case decryption becomes the following:

a) User A computes

$$d = \chi\delta^{-1}(C_1),$$

b) then computes

$$dC_2d = m. \tag{12.79}$$

The chosen ciphertext assault for this situation won't be fruitful, since the matrices X and m in general do not commute. An assault with a chosen ciphertext breaks cryptosystems RSA, Elgamal and Rabin, yet endeavors to assemble their alterations impervious to this assault, still brought about an extremely wasteful cryptosystems. As should be obvious, for the matrix modular cryptosystems circumstance is distinctive, since the closed variation varies from the normal just a couple of number of matrix multiplications [184]-[195].

12.5 Conclusion

In this chapter, we have proposed some new public-key cryptosystem with less computations and good security scheme. We have presented two different public key cryptosystems and tested both of these techniques with the help of numerical examples. Both cryptosystems are faster and balanced with respect to a pair of security-efficiency. In the most important case for the use of public-key cryptosystems, namely, key exchange protocols for symmetric ciphers. This modification can easily be used in different application due to its simplicity and efficiency with respect to security point of view.

Chapter 13

Conclusions

In literature and in life we ultimately pursue, not conclusions, but beginnings. Therefore, to conclude this thesis, a summary of the research performed is presented in the first section of this chapter which is followed by the discussion of possible future directions that could extend this research as mentioned in the final section of this chapter.

13.1 Summary of Thesis

The primary goal of the research work reported here in this thesis was to construct new techniques for creating S-boxes which optimized the properties of information security schemes that include cryptography, watermarking and steganography. The necessary background theory pertaining to Boolean functions, S-boxes, information security systems, and relevant prior research work performed by other researchers, has been outlined in the preceding chapters.

In this thesis, we have proposed four new techniques to be used for the improvement of cryptographically secure S-boxes. Each of the four techniques has been developed not only to focus on different significant strengthen properties, but also designed to optimize a combination of properties. These methods are novel, flexible and elegant, and were all successful in achieving their respective intended outcomes effectively.

The first technique for the construction of S-boxes is based on linear fractional transformation along with chaotic Lorenz systems. The output from chaotic systems is combining with constants of linear fractional transformation in order to produce numbers of S-boxes. These S-boxes satisfied significant cryptographic properties which include nonlinearity, Strict avalanche criteria, Bit independent criteria, Linear approximation probability, Differential approximation probability and compared the results with already existing well-known S-boxes. Also the extensions of these types of S-boxes along with Hyperchaotic Lorenz systems were seen in chapter 8 along with application to CAPTCHA.

The second technique of this thesis is based on chaotic Boolean functions. We have not only designed new S-box but also used it in image encryption applications. To verify the certifications of proposed chaotic S-box, we have performed standard analyses which include histogram, information entropy, correlation and, differential analyses which comprises of mean absolute (MAE), number of pixels change rate (NPCR) and unified average changing intensity (UACI). The values of all these coefficients are within the optimal ranges which is the confirmation of utility of suggested schemes.

The third scheme of the present thesis fundamentally consists of prime field of characteristic p, and is denoted $\mathbb{Z}_p$ where p is prime number. We have designed a new mechanism of S-boxes based on $\mathbb{Z}_{257}$ and $\mathbb{Z}_{17}$ along with S_4 permutations. The first mechanism is for image encryption which is described in chapter 6 and second one for steganography discussed in chapter 11.

The fourth designed technique is based on finite Galois rings of two types discussed in chapter 10. This chapter is specially designed in order to replace Galois field which is extensively used due to its vast applicability with the utility of invertibility. We have used maximal cyclic subgroup of Galois rings along with symmetry group S_n to generate large number of S-boxes. For the selection of particular S-box, we have applied chaotic iterative maps. Additionally, we have added very healthy applications of constructed S-boxes in watermarking and image encryption. The statistical analyses conducted for image encryption and watermarking clearly reflect the validity of suggested schemes.

Lastly, we have constructed a simple and effective public key algorithm which is based on the small Abelian subgroup of general linear group. Our scheme is fundamentally based on units of residue classes.

13.2 Future Directions

During and subsequent to the research performed for this thesis, a number of areas of future work have been identified. We now discuss directions for future research which involve both an extension of some of the work contained in this thesis, as well as topics of related work which could be investigated.

The basic idea of chapter 4 can be stretched for two dimensional linear fractional transformation along with six constants values which increases the number of S-boxes. As there is always a corner of improvement, so some new set of algebraic analyses can also be added to testify the strength of proposed S-boxes. Moreover, we can apply different chaotic relations to add randomness in our proposed chaos based S-boxes as discussed in chapter 7.

The constructions which are based on finite field of prime order and finite Galois rings (given in chapters 5,6,10,11) can be extended for audio and video encryption, watermarking and steganography. These construction techniques are completely different and novel. In chapter 12, we have considered public key encryption, which is based on units of modulo classes. The projected techniques can be extended to text and image encryption applications, which is another strength of this thesis.

Bibliography

[1] D. Machale, George Boole: his life and work (profiles of genius series), Boole Press, 1985.

[2] J. J. O'Connor, E. F. Robertson, George Boole, http://www.history.mcs.st-andrews.ac.uk/Biographies/Boole.html

[3] T. W. Cusick, P. Stanica, Cryptographic Boolean functions and applications, Elsevier/Academic Press, Amsterdam, 2009.

[4] C. E. Shannon, A mathematical theory of communication, Bell Labs. Tech. J., 27 (1948) 379–423.

[5] C. E. Shannon, Communication theory of secrecy systems, Bell Labs. Tech. J., 28 (1949) 656–715.

[6] N. J. A. Sloane, A. D. Wyner, Claude Elwood Shannon: collected papers, IEEE Press; 1993.

[7] P. Camion, C. Carlet, P. Charpin and N. Sendrier, On correlation-immune functions, Lect. Notes. Comput. Sc., 576 (1992) 86-100.

[8] C. Carlet, Boolean Functions for Cryptography and Error Correcting Codes, Chapter of the monography "Boolean Models and Methods in Mathematics, Computer Science, and Engineering", Cambridge University Press, (2010) 257-397.

[9] C. Carlet, Vectorial boolean functions for cryptography, Boolean Methods and Models, Cambridge Univ. Press, Cambridge.

[10] C. Carlet, Partially-bent functions, Design Codes Cryptogr., 3(2) (1993) 135-145.

[11] C. Carlet, On the propagation criterion of degree l and order k, Lect. Notes. Comput. Sc., 1403 (1998) 462–474.

[12] C. Carlet, A larger class of cryptographic Boolean functions via a study of the Maiorana-McFarland construction, Lect. Notes. Comput. Sc., 2442 (2002) 549-564.

[13] F. Chabaud, S. Vaudenay, Links between differential and linear cryptoanalysis, Lect. Notes. Comput. Sc., 950 (1995) 356-365.

[14] S. Chee, S. Lee, D. Lee, and S. H. Sung, On the correlation immune functions and their nonlinearity, Lect. Notes. Comput. Sc. , 1163 (1996) 232-243.

[15] B. Chor, O. Goldreich, J. Hastad, J. Freidmann, S. Rudich, and R. Smolensky, The bit extraction problem or $t-$resilient functions, In Foundations of Computer Science, 1985, 26th Annual Symposium, 396-407.

[16] T. W. Cusick, P. Stanica, Cryptographic Boolean functions and applications, Elsevier/Academic Press, Amsterdam, 2009.

[17] J. F. Dillon, A survey of bent functions, The NSA technical journal, (1972) 191-215.

[18] J. F. Dillon, Elementary Hadamard difference sets, PhD thesis, 1974.

[19] P. Duvall, J. Mortick, Some symptoms of boolean functions, 1970.

[20] R. Forre, The strict avalanche criterion: Spectral properties of boolean functions and an extended definition, Lect. Notes. Comput. Sc., 403 (1990) 450-468.

[21] K. Gopalakrishnan, D. R. Stinson, A short proof of the non-existence of certain cryptographic functions, J. Combin. Math. Combin. Comput., 20 (1996) 129-137.

[22] R. Lechner. Harmonic analysis of switching functions. In: Recent developments in switching theory, Academic Press, New York, 1971.

[23] R. Lidl, H. Niederreiter, Finite fields, Cambridge University Press, Cambridge, Second edition, 1997.

[24] S. Lloyd, Counting functions satisfying a higher order strict avalanche criterion, Lect. Notes. Comput. Sc., 434 (1990) 63-74.

[25] S. Lloyd, Balance, uncorrelatedness and the strict avalanche criterion, Discrete Appl. Math., 41(3) (1993) 223-233.

[26] F. J. MacWilliams, N. J. A. Sloane, The theory of error-correcting codes. II. North-Holland Publishing Co., Amsterdam, 1977. North-Holland Mathematical Library, Vol. 16.

[27] J. Maiorana, A class of bent functions, R41 Technical paper, 1970.

[28] H. B. Mann, Difference sets in elemantary abelian groups, Illinois J. Math., 9 (1965) 212-219.

[29] M. Matsui, Linear cryptoanalysis method for DES cipher, Lect. Notes. Comput. Sc., 765 (1994) 386-397.

[30] R. L. McFarland, A discrete fourier theory for binary functions, R41 Technical paper, 1971.

[31] W. Meier and O. Staelbach, Nonlinearity criteria for cryptographic functions, Lect. Notes. Comput. Sc., 434 (1990) 549-562.

[32] Q. Meng, H. Zhang, M. Yang, and J. Cui, On the degree of homogeneous bent functions, Discrete Appl. Math., 155(5) (2007) 665-669.

[33] P. K. Menon, On difference sets whose parameters satisfy a certain relation, Proc. Amer. Math. Soc., 13 (1962) 739-745.

[34] C. J. Mitchell, Enumerating boolean functions of cryptographic significance, J. Cryptology., 2(3) (1990) 155-170.

[35] K. Nyberg, Perfect nonlinear S-boxes, Lect. Notes. Comput. Sc., 547 (1991) 378-386.

[36] K. Nyberg, Differentially uniform mappings for cryptography, Lect. Notes. Comput. Sc., 765 (1994) 55-64.

[37] K. Nyberg, On the construction of highly nonlinear permutations, Lect. Notes. Comput. Sc., 658 (1993) 92-98.

[38] J. Pieprzyk, G. Finkelstein, Towards effective nonlinear cryptosystem design, IEEE Proc. Part E., 35(6) (1988) 325-335.

[39] Preneel, V. Leekwijk, V. Linden, Govaerts and Vandewalle, Propagation characteristics of boolean functions, Lect. Notes. Comput. Sc., 473 (1991) 161-173.

[40] B. Preneel, R. Govaerts and J. Vandewalle, Boolean functions satisfying higher order propagation criteria, Lect. Notes. Comput. Sc., 547 (1991) 141-152.

[41] O. S. Rothaus, On bent functions, J. Combin. Theory Ser. A., 20 (1976) 300-305.

[42] P. Sarkar, A note on the spectral characterization of correlation immune boolean functions, Inform. Process. Lett., 74 (2000) 191-195.

[43] J. Seberry, X. M. Zhang and Y. Zheng, Nonlinearity and propagation characteristics of balanced Boolean functions, Inform. Comput., 119(1) (1995) 1-13.

[44] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, IEEE Trans. Inf. Theory., 30(5) (1984) 776-780.

[45] G. J. Simmons, Contemporary cryptology, IEEE Press, New York, 1992. The science of information integrity.

[46] Y. Tarannikov, On resilient boolean functions with maximal possible nonlinearity, Lect. Notes. Comput. Sc., 2355 (2002) 66-77.

[47] W. D.Wallis, A. P. Street and J. S.Wallis, Combinatorics: Root squares, sum-free sets, Hadamard matrices, Lect. Notes. Math., 292 (1972) 160-161.

[48] A. F. Webster and S. E. Tavares. On the design of S-boxes, Lect. Notes. Comput. Sc., 218 (1986) 523-534.

[49] T. Xia, J. Seberry, J. Pieprzyk and C. Charnes, Homogeneous bent functions of degree n in $2n$ variables do not exist for $n > 3$, Discrete Appl. Math., 142(1-3) (2004) 127-132.

[50] G. Z. Xiao, J. L. Massey, A spectral characterization of correlation-immune combining functions, IEEE Trans. Inf. Theory., 34(3) (1988) 569-571.

[51] Y. Zheng and X. M. Zhang, Improved upper bound on the nonlinearity of high order correlation immune functions, Lect. Notes. Comput. Sc., 2012 (2001) 262-274.

[52] Y. Zheng and X. M. Zhang, On relationship among avalanche, nonlinearity, and correlation immunity, Lect. Notes. Comput. Sc., 1976 (2000) 470-482.

[53] Y. Zheng and X. M. Zhang, On plateaued functions, IEEE Trans. Inf. Theory., 17(3) (2001) 1215-1223.

[54] J. Christian, M. Hortmann and G. Leander, Boolean Functions, PhD thesis, (2012).

[55] W. Meier, O. Stafelbach, Nonlinearity criteria for cryptographic functions, Lect. Notes. Comput. Sc., 434 (1990) 549-562.

[56] R. L. McFarland, A family of difference sets in non-cyclic groups, J. Combin. Theory Ser. A., 15(1) (1973) 1-10.

[57] K. Nyberg, Constructions of bent functions and difference sets, Lect. Notes. Comput. Sc., 473 (1991) 151-160.

[58] J. Seberry, X. M. Zhang, Constructions of bent functions from two known bent functions, Australas. J. Combin., 9 (1994) 21-35.

[59] H. Dobbertin, Construction of bent functions and balanced Boolean functions with high nonlinearity, Lect. Notes. Comput. Sc., 1008 (1994) 61-74.

[60] C. Carlet, A construction of bent functions, Finite Fields Th. App. 233 (4) (1996) 47-58.

[61] S. Chee, S. Lee and K. Kim, Semi-bent functions, Lect. Notes. Comput. Sc., 917 (1995) 107-118.

[62] Y. Zheng, X. M. Zhang, Plateaued functions, Lect. Notes. Comput. Sc., 1726 (1999) 284-300.

[63] C. Carlet, E. Prouf, On plateaued functions and their constructions, Lect. Notes. Comput. Sc., 2887 (2003) 54-73.

[64] Y. Zheng, M. Zhang, Improved upper bound on the nonlinearity of high order correlation immune functions, Lect. Notes. Comput. Sc., 2012 (2001) 262-274.

[65] M. Matsui, The first experimental cryptanalysis of the Data Encryption Standard, Lect. Notes. Comput. Sc., 839 (1994) 1–11.

[66] H. M. Heys, A tutorial on linear and differential cryptanalysis, Technical Report CORR 2001-17, Centre for Applied Cryptographic Research, Department of Combinatorics and Optimization, University of Waterloo, March 2001.

[67] E. Biham, A. Shamir, Differential cryptanalysis of DES like cryptosystems, Lect. Notes. Comput. Sc., 537 (1991) 2–21.

[68] L. L. Bartosov, Linear and differential cryptanalysis of reduced-round AES, Tatra Mt. Math. Publ., 50(1) (2011) 51-61.

[69] E. Aras, M. D. Yucel, Performance Evaluation of Safer $K - 64$ and S-Boxes of the Safer Family, Turk. J. Electr. Eng. Co., 9(2) (2001) 161-175.

[70] M. Kontak, J. Szmidt, Nonlinearity of the round function, Control Cybern., 36(4) (2007) 1037–1044.

[71] A. Menezes, P. van Oorschot, S. Vanstone, Applied Cryptography. CRC, Boca Raton, 1996.

[72] A. Piva, F. Bartolini, M. Barni , Managing copyright in open networks, IEEE Trans. Internet Comput., 6(3) (2002) 18-26.

[73] C. Lu, S. Huang, C. Sze, H. Y. M. Liao , Cocktail watermarking for digital image protection, IEEE Trans. Multimedia., 2(4) (2000) 209-224.

[74] M. M. Latha, G. M. Pillai, K. A. Sheela, Watermarking based content Security and Multimedia Indexing in digital Libraries, International Conference on Semantic Web and Digital Libraries, (2007).

[75] W. Bender, D. Gruhi, N. Morimota, A. Lu, Techniques for Data Hiding, IBM. Syst. J., 35 (3-4) (1996) 313 - 336.

[76] A. Kejariwal, Watermarking, IEEE Potential, October/November, 2003, 37-40.

[77] J. Xuehua, Digital Watermarking and Its Application in Image Copyright Protection, 2010 International Conference on Intelligent Computation Technology and Automation, 2 (2010) 114 - 117.

[78] E. Feig, Fast algorithm for Discrete cosine transform, IEEE Trans. Signal Process., 40(9) (1992) 2174-2193.

[79] B. Pfitzmann, Information hiding terminology results of an informal plenary meeting and additional proposals, Lect. Notes. Comput. Sc., 1174 (1996) 347-350.

[80] R. Yadav, Study of Information Hiding Techniques and their Counterattacks: A Review Article, Int. J. Comput. Sci. Commun. Net., 1(2) (2011) 142-164.

[81] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn, Information Hiding A Survey, Proceedings of the IEEE, special issue on protection of multimedia content, 87(7) (1989) 1062-1078.

[82] A. Agarwal, Security Enhancement Scheme for Image Steganography using S-DES Technique, Int. J. Adv. Res. Comput. Sci. Soft. Eng., 2(4) (2012) 164-169.

[83] A. Almohammad, Steganography-Based Secret and Reliable Communications: Improving Steganographic Capacity and Imperceptibility, A thesis submitted for the degree of Doctor of Philosophy, Department of Information Systems and Computing , Brunel University, August, 2010.

[84] N. Johnson, S. Jajodia, Exploring steganography: seeing the unseen, IEEE Comput., 31 (1998) 26-34.

[85] E. Lin, E, Delp, A Review of Data Hiding in Digital Images, Proc. Image Process., 99 (1999) 25-28.

[86] N. F. Johnson, S. Katzenbeisser, A survey of steganographic techniques, Information Hiding, Artech House, (2000) 43-78.

[87] H. S. M. Reddy, K. B. Raja, High capacity and security steganography using discrete wavelet transform, Int. J. Comput. Sci. Sec., 3 (2009) 462-472.

[88] S. C. Katzenbeisser, Principles of Steganography, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, (2000) 43-78.

[89] P. Kruus, C. Scace, M. Heyman, M. Mundy, A survey of steganography techniques for image files, Adv. Sec. Res. J., 5(1) (2003) 41-52.

[90] M. S. Baptista, Cryptography with chaos, Phys. Lett. A., 240(1-2) (1998) 50-54.

[91] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a discrete chaotic cryptosystem using external key, Phys. Lett. A., 319(3-4) (2003) 334-339.

[92] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a chaotic secure communication system, Phys. Lett. A., 306(4) (2003) 200-205.

[93] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of dynamic look-up table based chaotic cryptosystems, Phys. Lett. A., 326(3-4) (2004) 211-218.

[94] S. Li, X. Mou, Z. Ji, J. Zhang, Y. Cai, Improving security of a chaotic encryption approach, Phys. Lett. A., 290(3-4) (2001) 127-160.

[95] G. Chen, Y. Chen, X. Liao, An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps, Chaos Solitons Fract., 31(3) (2007) 571–577.

[96] F. Özkaynak, A. B. Özer, A method for designing strong S-Boxes based on chaotic Lorenz system, Phys. Lett. A., 374(36) (2010) 3733–3738.

[97] Y. Wang, K. W. Wong, X. Liao, T. Xiang, A block cipher with dynamic S-boxes based on tent map, Commun. Nonlinear Sci. Numer. Simul., 14(7) (2009) 3089-3099.

[98] Y. G. Chen, X. Liao, An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps, Chaos Solitons Fract., 31(3) (2007) 571-577.

[99] T. Guoping, L. Xiaofeng, C. Yong, A novel method for designing S-boxes based on chaotic maps, Chaos Solitons Fract., 23(2) (2005) 413-419.

[100] G. Jakimoski, L. Kocarev, Chaos and cryptography: Block encryption ciphers based on chaotic maps, IEEE Trans. Circuits Syst., 48(2) (2001) 163-170.

[101] C. Adams, S. Tavares, Good S-boxes are easy to find, Lect. Notes. Comput. Sc., 89 (1989) 612–615.

[102] A. F. Webster, S. Tavares, On the design of S-boxes, Lect. Notes. Comput. Sc., 85 (1986) 523–534.

[103] J. Detombe, S. Tavares, Constructing large cryptographically strong S-boxes, Lect. Notes. Comput. Sc., 718 (1993) 165–181.

[104] M. Dawson, S. Tavares, An Expanded Set of S-Box Design Criteria Based on Information Theory and its Relation to Differential-Like Attacks, Lect. Notes. Comput. Sc., 91 (1991) 352–367.

[105] J. Pieprzyk, G. Finkelsten, Towards effective nonlinear cryptosystem design, IEEE Proc. Comput. Dig. Tech., 135(6) (1988) 325–335.

[106] I. Hussain, T. Shah, H. Mahmood, M. A. Gondal, A projective general linear group based algorithm for the construction of substitution box for block ciphers, Neural Comput. Appl., 22(6) (2013) 1085-1093.

[107] T. Shah, I. Hussain, M. A. Gondal, H. Mahmood, Statistical analysis of S-box in image encryption applications based on majority logic criterion, Int. J. Phys. Sci., 6(16) (2011) 4110-4127.

[108] A. Anees, Z. Ahmed, A Technique for Designing Substitution Box Based on Van der Pol Oscillator, Wireless Pers. Commun., (2015), DOI: 10.1007/s11277-015-2295-4.

[109] J. Zhang, D. X. Fang, H. Ren, Image Encryption Algorithm Based on DNA Encoding and Chaotic Maps, Math. Probl. Eng., 2014 (2014) 1-10.

[110] A. Jolfaei, A. Mirghadri, Survey: Image Encryption Using Salsa20, Int. J. Comput. Sci. Iss., 7(5) (2010) 213-220.

[111] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, Int. J. Bifurcation Chaos., 16(8) (2006) 2129–2153.

[112] G. Tang, X. Liao, Y. Chen, A novel method for designing S-boxes based on chaotic maps, Chaos Solitons Fract., 23(2) (2005) 413–419.

[113] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, An efficient method for the construction of block cipher with multi-chaotic systems, Nonlinear Dynam., 71(3) (2013) 489–492.

[114] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, An efficient technique for the construction of substitution box with chaotic partial differential equation, Nonlinear Dynam., 73(3) (2013) 1795-1801.

[115] M. Khan, T. Shah, A construction of novel chaos base nonlinear component of block cipher, Nonlinear Dynam., 76(1) (2014) 377–382.

[116] M. Khan, T. Shah, An efficient construction of substitution box with fractional chaotic system, Sig. Image Video Process., (2013), DOI 10.1007/s11760-013-0577-4.

[117] M. Khan, T. Shah, A Novel Statistical Analysis of Chaotic S-box in Image Encryption, 3D Res., 5(3) (2014) 1-8.

[118] M. Khan, T. Shah, S. I. Batool, Texture analysis of chaotic coupled map lattices based image encryption algorithm, 3D Res., 5(3) (2014) 1-19.

[119] M. Khan, T. Shah, A literature reviews on image encryption, 3D Research, 5(4) (2014) 1-25.

[120] Y. Wang, Q. Xie, Y. Wu, A software for S-box performance analysis and test, International Conference on Electronic Commerce and Business Intelligence, Beijing China, (2009) 125–128.

[121] N. Courtois, J. Pieprzyk, Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, Lect. Notes. Comput. Sc., 2501 (2002) 267-287.

[122] N. Courtois, Fast algebraic attacks on stream ciphers with linear feedback, Lect. Notes. Comput. Sc., 2729 (2003) 176-194.

[123] W. Meier, E. Pasalic, C. Carlet, Algebraic attacks and decomposition of Boolean functions, Lect. Notes. Comput. Sc., 3027 (2004) 474-491.

[124] P. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems, Lect. Notes. Comput. Sc., 1109 (1996) 104-113.

[125] E. Prouff, DPA Attacks And S-Boxes, Lect. Notes. Comput. Sc., 3557 (2005) 424-441.

[126] M. Sahar, M. E. Amir, Color image encryption based on coupled nonlinear chaotic map, Chaos Solitons Fract., 42(3) (2009) 1745–1754.

[127] H. J. Liu, X. Y. Wang, Color image encryption based on one-time keys and robust chaotic maps, Comput. Math. Appl., 59(10) (2010) 3320–3327.

[128] A. H. Qais, A. A. Aouda, Image Encryption Based on the General Approach for Multiple Chaotic Systems, J. Signal Inf. Process., 2(3) (2011) 238-244.

[129] S. Lian, J. Sun, Z. Wang, A block cipher based on a suitable use of the chaotic standard map, Chaos Solitons Fract., 26(1) (2005) 117-129.

[130] Q. Zhang, L. Guo, X. Wei, Image encryption using DNA addition combining with chaotic maps, Math. Comput. Model., 52(11-12) (2010) 2028-2035.

[131] S. E. Borujeni, M. Eshghi, Chaotic image encryption system using phase-magnitude transformation and pixel substitution, J. Telecommun. Sys., 52(2) (2013) 15-27.

[132] C. Zhu, A novel image encryption scheme based on improved hyperchaotic sequences, J. Optics Commun., 285(1) (2012) 29–37.

[133] G. Zhang, Q. Liu, A novel image encryption method based on total shuffling scheme, J. Optics Commun., 284(12) (2011) 2775–2780.

[134] S. Mazloom, A. M. Eftekhari-Moghadam, Color image encryption based on coupled nonlinear chaotic map, Chaos Solitons Fract., 42(3) (2009) 1745–1754.

[135] H. S. Kwok, W. K. S. Tang, A fast image encryption system based on chaotic maps with finite precision representation, Chaos Solitons Fract., 32(4) (2007) 1518–1529.

[136] N. Ferguson, R. Schroeppel, D.Whiting, A simple algebraic representation of Rijndael, Lect. Notes. Comput. Sc., 2259 (2001) 103–111.

[137] N. Mentens, L. Batina, B. Preneel, I. Verbauwhede, A systematic evaluation of compact hardware implementations for the Rijndael S-box, Lect. Notes. Comput. Sc. , 3376 (2005) 323–333.

[138] H. Khanzadi , M. A. Omam, F. Lotfifar, M. Eshghi, Image encryption based on gyrator transform using chaotic maps, IEEE Conference on Signal Processing, (2010) 2608–2612.

[139] T. Baigèneres, Y. Lu, S. Vaudenay, P. Junod, J. Monnerat, A Classical Introduction to Cryptography Exercise Book, Springer US, 2005.

[140] M. Hénon, A two-dimensional mapping with a strange attractor, Commun. Math. Phys., 50(1) (1976) 69–77.

[141] W. F. H. Al-Shameri, Dynamical properties of the Hénon mapping, Int. J. Math. Anal., 6(49) (2012) 2419–2430.

[142] H. K. Sarmah, R. Paul, Period doubling route to chaos in a two parameter invertible map with constant Jacobian, Int. J. Res. Rev. Appl. Sci., 3(1) (2010) 72–82.

[143] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Sys. Man and Cyber., 3(3) (1973) 610–621.

[144] J. M. H. Buf, M. Kardan, M. Spann, Texture feature performance for image segmentation, Pattern Recognit., 23(3-4) (1990) 291–309.

[145] J. F. Haddon, J. F. Boyce, Co-occurrence matrices for image analysis, Elect. Commun. Eng. J., 5(2) (1993) 71–83.

[146] P. P. Ohanian, R. C. Dubes, Performance evaluation for four class of texture features, Pattern Recognit., 25(8) (1992) 819–833.

[147] R. M. Haralick, Statistical and structural approaches to texture, Proc. IEEE., 67 (1979) 786–804.

[148] M. R. May, Simple mathematical models with very complicated dynamics, Nature,, 261(1976) 459-467.

[149] V. P. François, Notice sur la loi que la population poursuit dans son accroissement, Correspondance mathématique et physique, 10 (2013) 113–121.

[150] L. V. Ahn, M. Blum, N. J. Hopper, J. Langford, The CAPTCHA, Web Page: http://www.captcha.net. 2000.

[151] L. V. Ahn, M. Blum, J. Langford, Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI, Communications of the ACM, 47 (2004) 57-60.

[152] M. Bellare, R. Impagliazzo, M. Naor, Does Parallel Repetition Lower the Error in Computationally Sound Protocols?, IEEE Symposium on Foundations of Computer Science, 97 (1997) 374-383.

[153] M. M. Bongard, Pattern Recognition, Spartan Books, Rochelle Park NJ, 1970.

[154] A. L. Coates, H. S. Baird, R. J. Fateman, Pessimal Print: A Reverse Turing Test, Proceedings of the International Conference on Document Analysis and Recognition, (2001) 1154-1159.

[155] S. Craver, On Public-key Steganography in the Presence of an Active Warden, Lect. Notes. Comput. Sc., 1525 (1998) 355-368.

[156] N. J. Hopper, J. Langford, L. V. Ahn, Provably Secure Steganography, Lect. Notes. Comput. Sc., 2442 (2002) 77-92.

[157] M. D. Lillibridge, M. Abadi, K. Bharat, A. Broder, Method for selectively restricting access to computer systems, US Patent 6 (2001).

[158] G. Mori, J. Malik, Breaking a Visual CAPTCHA, Unpublished Manuscript, 2002, Available electronically: http://www.cs.berkeley.edu/~mori/gimpy/gimpy.pdf.

[159] M. Naor, Verification of a human in the loop or Identification via the Turing Test, Unpublished Manuscript, 1997, Available electronically: http://www.wisdom.weizmann.ac.il/~naor/ PAPERS/human.ps.

[160] B. Pinkas, T. Sander, Securing Passwords Against Dictionary Attacks, Proceedings of the ACM Computer and Security Conference, 2 (2002) 161-170.

[161] S. Rice, G. Nagy, T. Nartker, Optical Character Recognition: An Illustrated Guide to the Frontier, Kluwer Academic Publishers, Boston, 1999.

[162] A. Shamir, E. Tromer, Factoring Large Numbers with the TWIRL Device, Unpublished Manuscript, 2003, Available electronically: http://www.cryptome.org/twirl.ps.gz.

[163] J. Xu, R. Lipton, I. Essa, Hello, are you human, Technical Report GIT-CC-00-28, Georgia Institute of Technology, November 2000.

[164] J. P. Pieprzyk, Non-linearity of Exponent Permutations, Lect. Notes. Comput. Sc., 434 (1990) 80-92.

[165] S. I. Batool, T. Shah, M. Khan, A color image watermarking scheme based on affine transformation and S_4 permutation, Neural Comput. Appl., 25(7-8) (2014) 2037-2045.

[166] S. D. Sinha, C. P. Arya, Algebraic Construction and Cryptographic Properties of Rijndael Substitution Box, Defence Sci. J., 62(1) (2012) 32-37.

[167] M. Kontak, J. Szmidt, Nonlinearity of the round function, Control Cyber., 36(4) (2007) 1037-1044.

[168] R. Coulter, M. Henderson, R. Matthews, A note on constructing permutation polynomials, Finite Fields Th. App., 15(5) (2009) 553-557.

[169] A. Akbarya, D. Ghiocab, Q. Wang, On constructing permutations of finite fields, Finite Fields Th. App., 17(1) (2011) 51–67.

[170] S. Gao, J. V. Z. Gathen, D. Panario, V. Shoup, Algorithms for Exponentiation in Finite Fields, J. Symb. Comput., 29(1) (2000) 879–889.

[171] L. Wang, Y. Zhang, J. Feng, On the Euclidean Distance of Images, IEEE Trans. Pattern Analy. Mach. Intel., 27(8) (2005) 1334 - 1339.

[172] J. L. Falcone, P. Albuquerque, A correlation-based distance, Available Electronically: http://www.arxiv.org/abs/cs.IR/0402061, February 2004.

[173] B. Salton, S. Gerard, B. Christopher, Term-weighting approaches in automatic text retrieval, Inform. Process. Manag., 24(5) (1998) 513–523.

[174] C. E. Shannon, W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, Urbana, IL, 1949.

[175] T. M. Cover and J.A. Thomas, Elements of Information Theory, Wiley, New York 1991.

[176] R. C. Hilborn, Chaos and Nonlinear Dynam: an introduction for scientists and engineers, Second Edition, Oxford, University Press, New York, 2004.

[177] A. D. Ker, Steganalysis of LSB matching in grayscale images, IEEE Sig. Process. Lett., 12(6) (2005) 441-444.

[178] C. Chang, T. D. Kieu, A reversible data hiding scheme using complementary embedding strategy, Inform. Sci., 180(16) (2010) 3045-3058.

[179] T. Jamil, Steganography: The art of hiding information is plain sight, IEEE Poten., 18(1) (1999) 10-12.

[180] C. K. Chan, L. M. Chang, Hiding data in image by simple LSB substitution, Pattern Recognit., 37(3) (2003) 469-471.

[181] R. Kaur, B. Singh, I. Singh, A Comparative Study of Combination of Different Bit Positions In Image Steganography, Int. J. Mod. Eng. Res., 2(5) (2012) 3835-3840.

[182] J. Brenard, Digital Image Processing, Springer-Verlag Berlin Heidelberg, 2005.

[183] S. S. Agaian, B. Silver, K. A. Panetta, Transform coefficient histogram-based image enhancement algorithms using contrast entropy, IEEE Trans. Image Process., 16(3) (2007) 741-758.

[184] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory., 22(6) (1976) 644-654.

[185] R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Commun. ACM., 21(2) (1978) 120-126.

[186] M. O. Rabin, Digitized signatures and public-key functions as intractible as factorization, MIT Laboratory for Computer Science Technical Report, LCS/TR-212 (1979).

[187] H. C. Williams, A Modification of the RSA Public-Key Encryption Procedure, IEEE Trans. Inf. Theory., 26(6) (1980) 726-729.

[188] H. C. Williams, Some public-key crypto-functions as intractible as factorization, Lect. Notes. Comput. Sc., 196 (1985) 66-70.

[189] P. Smith, M. Lennon, A newpublic key system, Proceedings of the IFIP TC11 Ninth International Conference on Information Security, 37(12-14) (1993) 103-117.

[190] Z. Cao, Conic analog of RSA cryptosystem and some improved RSA cryptosystems, J. Nat. Sci. Heilongjiang Univ., 16(4) (1999) 5-18.

[191] Z. Cao, A threshold key escrow scheme based on public key cryptosystem, Sci. China (E Series)., 44(4) (2001) 441-448.

[192] K. Komaya, U. Maurer, T. Okamoto, S. Vanston, Newpublic-key schemes bases on elliptic curves over the ring $\mathbb{Z}_n$, Lect. Notes. Comput. Sc., 576 (1992) 252-266

[193] Z. Cao, The multi-dimension RSA and its low exponent security, Sci. China (E Series)., 43(4) (2000) 349-354.

[194] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inf. Theory., 31(4) (1985) 469-472.

[195] K. McCurley, A key distribution system equivalent to factoring, J. Cryptology., 1(2) (1988) 95-100.

[196] P. Shankar, On BCH codes over arbitrary integer rings, IEEE Trans. Inf. Theory., 25(4) (1979) 480-83.

[197] A. G. Shanbhag, P. V. Kumar, T. Helleseth, Upper bound for a hybrid sum over Galois rings with applications to aperiodic correlation of some $q-$ary sequences, IEEE Trans. Inf. Theory., 42(1) (1996) 250-54.

[198] A. A. Andrade, R. Palazzo, Construction and decoding of BCH codes over finite rings, Linear Algebra Appl., 286(1-3) (1999) 69-85.

[199] T. Shah, A. Qamar, I. Hussain, Substitution box on maximal cyclic subgroup of units of a Galois ring, Z. Naturforsch. A., 68a (8-9) (2013) 567-572.

[200] T. Abualrub, I. Saip, Cyclic coacquired a great consideration in algebraic coding theory over the rings $Z_2 + uZ_2$ and $Z_2 + uZ_2 + u^2Z_2$, Design Codes Cryptogr., 42(3) (2007) 273-287.

[201] M. A. Ashker, Simplex codes over the ring $\sum_{n=0}^{s} u^n F_2$, Turk. J. Math., 29(3) (2005) 221-233.

[202] M. A. Ashker, Simplex codes over $F_2 + uF_2$, Arab. J. Sci. Eng., 3(2A) (2005) 227-285.

[203] M. A. Ashker, M. Hamoudeh, Cyclic codes over $F_2 + uF_2 + \cdots + u^{k-1}F_2$, Turk. J. Math., 33(5) (2011) 737-749.

[204] M. A. Ashker, J. Chen, Cyclic codes of arbitrary length over $F_q + uF_q + \cdots + u^{k-1}F_q$, Palistine J. Math., 2(1) (2013) 72-80.

[205] B. Bilgin, S. Nikova, V. Nikov, V. Rijmen, G. Stutz, Thershold Implementations of all 3 × 3 and 4 × 4 S-boxes, Cryptographic Hardware and Embedded Systems, Springer Berlin Heidelberg, 2012, 76-91.

[206] A. Bonnecaze, P. Udaya, Cyclic codes and self dual codes over $F_2 + uF_2$, IEEE Trans. Inf., 45(4) (1999) 1250-1255.

[207] W. E. Clark, J. J. Liang, Enumeration of finite commutative chain rings, J. Algebra., 27(3) (1973) 445-453.

[208] X. Hou, Finite commutative chain rings, Finite Fields Th. App., 7(3) (2001) 382–396.

[209] A. Naji, Linear codes over $F_2 + uF_2 + u^2F_2$ of Constant Lee weight, The second conference of the Islamic University on Mathematical Science-Gaza, August, 2002.

[210] J. Qian, L. Zhang, S. Zhu, Cyclic codes over $F_p + uF_p + \cdots + u^{k-1}F_p$, IEICE Trans. Fundamentals., 3 (2005) 795-779.

[211] J. Qian, L. Zhang, S. Zhu, $(1+u)$ constacyclic and cyclic over $F_2 + uF_2$, Appl. Math. Lett., 19(8) (2006) 820-823.

[212] J. Qian, L. Zhang, S. Zhu, Constacyclic and cyclic codes over $F_2 + uF_2 + u^2F_2$, IEICE Trans. Fundamentals., 6 (2006) 1863-1885.

[213] T. Shah, A. Qamar, A. A. Andrade, Constructions and decoding of a sequence of BCH codes, Math. Sci. Res. J., 16(9) (2012) 234-250.

[214] T. Shah, A. Qamar, A. A. Andrade, Construction and decoding of BCH codes over chain of commutative rings, Math. Sci., 6(51) (2012) 1-14.