ANALYSIS AND SYNTHESIS OF FUNCTIONS OVER THE RINGS Z_q WITH NOVAL CRYPTOGRAPHIC CHARACTERISTICS

A thesis submitted to the **University of Calicut** for the award of the degree of

DOCTOR OF PHILOSOPHY

In

MATHEMATICS

By

ABOOBACKER P

Under the guidance of

Dr. VIJI M

Associate Professor,

Department of Mathematics,

ST. THOMAS COLLEGE (AUTONOMOUS),THRISSUR

KERALA - 680 001, INDIA

Research and Postgraduate Department of Mathematics

St. Thomas College (Autonomous)

Thrissur - 680001

June 2024

DECLARATION

I, **Aboobacker P**, hereby declare that the thesis titled **"Analysis and Synthesis of Functions over the Rings Z_q with Noval Cryptographic Characteristics "** submitted to the University of Calicut, Kerala in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy in Mathematics** is a record of original and independent research work done by me under the supervision of **Dr. Viji M, Associate Professor, St. Thomas College (Autonomous), Thrissur**. I also declare that this thesis or any part of it has not been submitted to any other University/Institute for the award of any degree.

Thrissur

15 June, 2024 Aboobacker P

CERTIFICATE

This is to certify that the thesis titled **"Analysis and Synthesis of Functions over the Rings Z_q with Noval Cryptographic Characteristics"** submitted by **Aboobacker P** to the University of Calicut in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy in Mathematics** is a record of original research work carried out by him under my supervision. The content of this thesis, in full or in parts, has not been submitted by any other candidate to any other University for the award of any degree or diploma.

Thrissur

15 June, 2024

Dr. Viji M

Associate Professor,

St.Thomas College (Autonomous)

Thrissur.

ABSTRACT

In the design of cryptographic algorithms, mathematical techniques, particularly those based on number theory, abstract algebra, and linear algebra, have an inevitable place. Symmetric cryptographic algorithms like block ciphers utilize finite field arithmetic and linear algebra for their construction. The public key cryptographic algorithms are sound with number-theoretical methods. Modern telecommunications technology is increasingly using advanced algebraic formulations. As computational power keeps growing quickly, new ideas and techniques for data encryption are being developed to safeguard information as it is transmitted over open channels. So developing more effective cryptographic primitives involving algebraic techniques is a task strongly tied to further improving present cryptographic information protection and communication systems.

The Boolean function is an important cryptographic primitive with which most of the modern block ciphers are built. However, the cryptanalytic methods are not constrained in the representation of these block ciphers. The modern block ciphers can also be represented using primitives on non-binary logic(many-valued logic) also. As a result, an attacker could employ many-valued logic based algebraic techniques to break the design. The employment of many-valued logic based algorithms gives users considerably more freedom in the selection of transformations than binary equivalents. Consequently, it is a fascinating area of research to analyse the cryptographic characteristics of the primitives based on many-valued logic in order to build algorithms based on them.

Through this thesis, the main focus is on the analysis of the cryptographic properties of the cryptographic primitives based on many-valued logic, the functions over the rings Z_q, giving more emphasis on functions over Z_3 (ternary functions) and functions over Z_4 (quaternary functions).

From here onwards, functions over the rings Z_q, q-functions and functions of q-valued logic are synonymous terms.

The chapters of the thesis are arranged as follows: The development of cryptography from ancient times to the present day is discussed in Chapter 1. This chapter also covers basic definitions of cryptography and the classification of cryptographic algorithms.

In Chapter 2, a detailed review of the cryptographic properties of Boolean functions that lead to the study of cryptographic characteristics of functions over Z_q is covered. The preliminaries required for the forthcoming chapters are also discussed in Chapter 2.

In Chapter 3, a method for spectral analysis of rotation symmetric functions of q-valued logic, that significantly reduces the computational complexity, in its Vilenkin-Chrestenson domain is discussed, and this method is utilised to synthesise the rotation symmetric ternary bent functions in three variables.

Chapter 4 discusses the correlation immunity and resiliency of functions over Z_q and presents a mathematical expression that generates the complete class of quaternary resilient functions in two variables. Orthogonal characterization of the resiliency of quaternary functions in terms of sub-functions is also presented. The construction of m-resilient functions of $n+1$ variables from that of n variables

is discussed.

Chapter 5 describes the design of a cryptographic primitive, known as substitution box (S-box), on ternary logic, and cryptographic properties such as nonlinearity, avalanche characteristics, and measures of input-output correlation are analysed.

Chapter 6 presents the concluding remark of the thesis and chapter 7 discusses the proposal for future work.

To my beloved family

ACKNOWLEDGEMENTS

"My Success is only by Lord" (Quran 11:88).

On the whole, I would like to humbly place on record, my deep sense of gratitude towards the God Almighty who stooped down and led me amazingly through the path of grace and wisdom.

I am thankful for the unwavering support and invaluable insights provided my supervisor Dr.Viji M, Assistant Professor and Head, Department of Mathematics, St. Thomas College (Autonomous),Thrissur, during the period of research.

I want to acknowledge and thank Mr.Vishnudas, the one who supported me in navigating the complexities of this academic journey. My heartfelt appreciation goes out to Dr.Artem Sokolov, Odessa National Polytechnic University, Ukraine, for the academic discussions. I wish to offer my profound thanks to the research colleagues of the Department of Mathematics, St.Thomas College (Autonomous), Thrissur.

I acknowledge St.Thomas College(Autonomous), Thrissur with gratitude for providing me with the resources and facilities required for the research work. I express my gratitude to the principal Dr. Fr. Martin K A, former principals Dr. Joy KL, and Dr. Ignaitius Antony, and all teaching and non-teaching staff of St.Thomas College(Autonomous), Thrissur. I would like to convey my sincere gratitude to former heads of the department Dr.Saju M I, Prof.Vincent Pulikkottil, and all faculties of the Department of Mathematics.

This thesis is incomplete without mentioning the support given by family, teachers, and friends. Their encouragement has been instrumental, and I am truly thankful. In conclusion, I am indebted to everyone who aided me in achieving this thesis milestone.

Aboobacker P

Contents

List of Figures

List of Tables

CHAPTER 1

General Introduction

1.1 The History of Cryptography

The human mind's inclination for concealment has been one of the strongest innate traits from the origin of mankind. Ever since humans emerged from the caves, started to live in societies, and decided to give civilization some real thought, we have felt the need to conceal our communications. The idea that we have to work together emerged and blossomed as soon as there were different tribes or groups, along with overt conflict, hiding, and crowd control. While we were kids, many of us had magic decoder rings that we used to communicate with friends using coded messages while may be withholding information from our parents, siblings, or teachers. There are numerous instances throughout history where people have attempted to conceal information from adversaries. During wartime, various communication strategies were used by kings and generals to keep the enemy from gathering vital military intelligence. The demand for increasingly advanced data protection techniques has grown as society has developed. The need

for information transmission and electronic services is rising as the globe gets more connected, and with that growth comes a greater reliance on electronic systems. It is already a normal practice to communicate sensitive information, such as credit card details, over the internet. Thus electronic systems and data security are essential to our way of life. Man created several strategies for the secure transmission of information, which later became a science known as cryptography.

It is not surprising that the very first evidence of the use of cryptography was found in the areas now occupied by Egypt, Greece, and Rome, the cradle of civilization. The first instances of cryptography are found in inscriptions in the main chamber of the tomb of the Egyptian aristocrat Khnumhotep II. Egyptians used 'hieroglyph', a cryptographic technique for communication. The scribe occasionally replaced uncommon hieroglyphic symbols throughout the message. The intention was not to hide the information; instead, it was to convey it in a way that made it appear more sophisticated than it actually was. The Greeks came up with an encryption technique known as scytale cipher, composed of a cylinder with a message written on a strip of parchment wrapped around it. The lettering on the cylinder would be meaningless when it was unwound. Naturally, the recipient of the message would have a cylinder of the same diameter and utilise it for message decoding. The Caesar Shift Cipher is a Roman cryptographic technique. Around 100 BC, it is known that Julius Caesar used an encryption technique which is the first known form of the modern cryptographic technique used to share secret information with his military officials deployed at the war front. It made use of the concept of composing the message using a letter shifting by a predetermined number. The most common number used was three. The intended recipient would then counter-shift the letters by the same number to read the message.

In 1553, a cryptographer from Italy, Giovan Battista Bellaso, described a text auto-key cipher that was thought to be unbreakable for four centuries. He proposed a method in which the alphabet is identified using a consented countersign or a keyword among the parties involved in communication. Additionally, he demonstrated various techniques for combining cipher alphabets to overcome the necessity for the correspondents to exchange discs or predetermined tables.

During the 16^{th} century, a method for encrypting alphabetic text was developed by Vigenere and is known as Vigenere Cipher. It employs a method of polyalphabetic substitution, a substitution-based encryption that employs several substitutions of alphabets at a time.

In 1917, an American named Edward Hebern designed an electro-mechanical device known as the Hebern rotor machine. It makes use of a single rotor in which a rotating disc contains the hidden key. Each key press on the keyboard generated cipher text since the key was encoding a substitution table. The disc was likewise rotated by one notch, as a result, a new table was then utilised for the subsequent character of plain text.

The Enigma machine was another encryption machine created in 1918 by German inventor Arthur Scherbius and used by the Germans to send coded messages during World War II. Because there are countless ways to code a message using an Enigma machine, it was extremely challenging for other countries to decipher German codes during World War II. It employs many rotors as opposed to Hebern's device's single rotor. By taking advantage of a few flaws in the Enigma code's implementation and gaining access to German code books, Alan Turing and other academics were able to create the Bombe machine, a device that assisted in cracking even the most difficult Enigma codes.

American mathematician, electrical engineer, and cryptographer Claude E. Shannon is considered as the ”father of information theory.” Shannon described information and how it can be communicated in terms of mathematics, in his classic work ”A Mathematical Theory of Communication,” which was written at Bell Labs in 1945 and later published as a book in 1949 under the title ”The Mathematical Theory of Communication”. Secrecy and authenticity were outlined by Shannon as the two primary objectives of cryptography. Shannon presented a study that demonstrated the possibility of an unbreakable cryptographic scheme a year after he introduced information theory.

The one-time pad or Vernam cipher is a cryptographic method, which was created by Gilbert Vernam close to the end of World War I. The goal is to make the message fully random by encoding it using a key made up of a random string of digits.

Significant developments occurred in the 1970s. A research team at IBM proposed and submitted the Data Encryption Standards (DES)[1] cipher on March 17, 1975, in response to a request from the National Bureau of Standards (NBS), now known as the National Institute of Standards and Technology (NIST), in an effort to create secure electronic communication tools for companies like banks and other large financial organizations. A slightly modified version was chosen by the NBS in 1976 after consultation with the National Security Agency (NSA), and it was made an official Federal Information Processing Standard (FIPS) for the United States in 1977. DES was the first encryption Scheme approved by the NSA that the public could access.

NIST announced a competition for the replacement of DES in January 1997. Following a five-year process of standardization, fifteen algorithms were presented and evaluated. In October 2000, the Rijndael algorithm

developed by Belgian cryptographers Joan Daemen and Vincent Rijmen was declared by NIST as the winner. The Rijndael block cipher variant named as the Advanced Encryption Standard (AES)[2] was declared the U.S. federal government standard by NIST in 2001.

1.2 Cryptology

We are currently in a technological age, and as technology develops, new approaches to solving problems related to secure communication are becoming more and more necessary. Cryptology is one of the oldest fields that explore numerous methods for secure information transfer.

The word ‘cryptology ‘originates from the Greek words Kryptos, which means concealed, and logos, which means speech. Cryptology comprises two major fields cryptography and cryptanalysis. The art of writing codes is known as cryptography, and the methods for cracking codes are known as cryptanalysis. In response to the work by Shannon that was published in 1949 [3], it gained popularity and was turned into a science.

During communication, the message to be communicated is changed into something that appears random and meaningless to shield it from being read by unauthorized persons. The message that needs to be shared is referred to as plaintext in the language of cryptography, and the encoded message is referred to as ciphertext. Encryption is the process of changing plaintext into ciphertext. The restoring process to retrieve the original content from the coded message is known as decryption.

Secure data transmission is the goal of cryptographic algorithms that meet the essential requirements of security protocols. Since the need for secure information sharing arose in antiquity when communication routes were still rudimentary and largely accessible to all opponents, cryptography is arguably one of the oldest fields. Cryptography is the study of mathemat-

ical methods related to information security aspects. The studies revolve around the four pillars confidentiality, data integrity, entity authentication, and non-repudiation. The service used to keep the content of the data safe from unauthorized entities is termed confidentiality. Data manipulation is one of the threats in the data transmission process. Data integrity guarantees that no adversary may tamper with or change the data. Authentication is the process of confirming the origin and integrity of the information. Non-repudiation is a service that prevents an entity from denying a previous commitment or action. Cryptanalysis is the technique that exploits Mathematical and technological advancements to break the algorithm. The tremendous rise in technology makes the algorithms insecure because of the various attacks by an adversary using modern techniques. Thus, there is a requirement for more potential tools for data communication. So design and analysis of cryptographic tools (primitives) is a field of study that will never end.

A few techniques that are adopted to achieve secrecy led to the development of various cryptographic algorithms. Symmetric key, asymmetric key, hash functions, and digital signatures could be considered as major classifications.

The first two classifications depend on the secrecy of a string of characters, known as a key that is connected to the encryption process. Symmetric key cryptosystems utilise a single key that is kept secret between communication entities and is used for both encryption and decryption. Asymmetric cryptographic algorithms have a pair of keys consisting of a private key that is kept secret and a public key that is known to everyone in the network. The Mathematical apparatus, one-way functions are utilised to generate the key pairs. Asymmetric cryptography is also known as public-key cryptography. RSA, Diffie-Hellman, and Elliptic Curve Cryptography (ECC) are a

few examples of asymmetric encryption techniques. These algorithms are rich with methods in number theory.

The symmetric cryptosystem is further classified into block ciphers and stream ciphers. Block ciphers encrypt an entire block of data, while stream ciphers only transform one unit at a time. The widely used ciphers like AES and DES are examples of block ciphers and examples of stream ciphers include RC4, Salsa20, PANAMA, etc.

Hash functions are used for data integrity and authentication. A hash function converts data of various lengths to fixed lengths known as a hash value or message digest. A hash value is a miniature form of the data to be communicated. The receiver can ensure the integrity of the data by checking this hash value.

A digital signature is also a mathematical technique used to validate the authenticity of the data. A digital signature is a method of linking a person or entity to digital information. This agreement can be independently confirmed by the receiver and any other party. Because they are cryptographically linked to the signed document and are verifiable, digital signatures are regarded as a more secure form of e-signature.

1.3 Cryptography and Finite Fields

Finite fields have a pivotal role in the construction of most of the popular cryptographic constructions[2, 4]. Since the data is represented as a sequence of bits consisting of 0's and 1's, the finite field of order 2^n constructed over the base field $F_2 = \{0, 1\}$ has significant importance in cryptography. These fields are also known as Galois Field denoted by $GF(2^n)$. The invertibility property of the non-zero elements in the finite field ensures the encryption-decryption processes, hence the representation of the digital data in a cryptographic construction in terms of elements

of a finite field is crucial. For example, in the block cipher AES, data is represented as a block of 128 bits consisting of 16 bytes. The major operations are performed over each byte. One may look at each byte as an element of the Galois field $GF(2^8)$ as well as a polynomial of degree at most 7, of the quotient ring $F_2[x]/(p(x)$ for properly chosen irreducible polynomial $p(x)$ of degree 8[2, 4].

There are various cryptographic primitives based on non-binary logic working on the principles of the Galois Field. For example, for analysing the cryptographic characteristics of quaternary functions, it is required to consider the algebraic operations over the Galois field $GF(2^2)$ [5].

1.4 Boolean Functions

The Boolean functions (BF) are the fundamental component of many cryptosystems, especially symmetric cryptographic systems. To design an efficient cryptographic system, a detailed study of the properties of a BF is essential. For example, the Substitution box, which is the major nonlinear component of the block ciphers is extracted from BFs[2]. The strength of such block ciphers depends on the strength S-box. So if the component Boolean functions are not chosen properly, it will adversely affect the security of the algorithm.

Thus, it is vital to construct BFs with significant cryptographic properties to design viable cryptographic algorithms using them as their components. Strict Avalanche criterion (SAC), nonlinearity (NL), algebraic immunity (AI), correlation immunity(CI), and Bit Independence Criterion(BIC) are a few of the characteristics that a Boolean function possesses for designing a secure cryptosystem.

1.5 Cryptography and Functions over the rings Z_q

Recently, studies on algorithms based on non-binary (many-valued) cryptographic primitives are of research interest. One of the main benefits of employing many-valued logic is the ability to optimize or minimize data processing, so it is important to take full advantage of the potential of many-valued logic. There are electronic devices in which models with more than two states (fuzzy logic) are effectively implemented. Many-valued logic is an alternating solution for many practical problems. It has applications in cryptography, coding theory, signal processing, VLSI ICs, etc. The application of functions of q-valued logic simplifies the design of position-based cryptographic schemes [6]. In signal processing, many-valued logic is convenient in the storage and processing of a huge amount of data encoded in a digital signal, and additionally, numerous strategies implemented in signal processing are geared up sufficiently to remedy particular issues confronted in the design of systems based on many-valued logic [7]. If the signals in an integrated circuit permit to accept four states (quaternary values) instead of only two states (binary values), the problem encountered in some VLSI circuits due to the constraints in the number of connections that can be used within the circuit and that connect the outer world can be solved effectively [8]. Some VLSI ICs are available intended for commercial purpose that has design principle based on quaternary logic.

There is potential for algorithms based on non-binary systems as quantum computation develops. One of the fundamental components of many-valued logic based cryptographic construction is functions over Z_q.In order to develop such an algorithm, it is necessary to analyse the cryptographic features of functions over Z_q that are analogous to BFs. Many-valued logic based approaches allow much greater flexibility in the selection of transformations than binary alternatives. Therefore, in the field of many-valued

logic, active research and developments are undergoing [9].

The choice of mathematical tool that the cryptanalyst uses to represent the cryptographic design is not restricted so that alternative representations are feasible for subsequent attacks. The representation via q-functions like 4-function, as well as 16-functions, could be utilised for the representation of existing modern ciphers, in addition to the Boolean functions employed in the design of the cryptographic algorithm. The designers of cryptographic algorithms typically do not consider these potential representations or explore their cryptographic merits. This scenario makes it necessary to construct, develop, and broaden the cryptographic quality standards and the practical implications of the functions over the rings Z_q [10].

1.6 Quantum computing and Post Quantum Cryptography

"Quantum computing represents a fundamental shift because it harnesses the properties of quantum mechanics and gives us the best chance of understanding the natural world," Mr. Pichai.

The notion of quantum computing was first put forth in the 1980s as a way to advance computational techniques by integrating concepts from quantum physics, classical information theory, and computer science, to solve mathematical problems that are difficult for classical computers[11]. This integration identifies the information as basic concepts in quantum physics. This significance is solidified by the notion of quantum information and computing, which has offered some significant and fascinating insights into reality. The popularity of the field was amplified by the introduction of Shores's algorithm[12] proposed by an American Mathematician, Peter Shor, that would speed up cryptanalysis and threaten some cryptographic algorithms used if implemented on a quantum computer.

Zeros and ones are used in traditional computers to store information. Quantum bits, also known as qubits, are the basic unit of information in a quantum computer. Qubits represent and store information in a quantum state that is a complex combination of zeros and ones. A single qubit isn't really useful. However, by putting the quantum data into a superposition state, which encapsulates all qubit configurations that might be possible, complex multidimensional computational spaces can be produced using qubit groups and these spaces allow for the addressing of complex problems. The spin orientation of the quantum particles like photons or electrons determines the state of the qubit.

The security of contemporary cryptographic algorithms like RSA, El Gamal, and Elliptic Curve Cryptography, which rely solely on the difficulty of factoring large numbers and computing discrete logarithms that could be effectively solved using Shore's algorithm, is questioned by the development of quantum computers[13]. So, developing methods for addressing this issue is necessary. The research leads to post-quantum cryptography. The aim of post-quantum cryptography, also known as quantum-resistant cryptography, is to create cryptosystems that are robust to both classical and quantum computers and can work with already-existing network topologies and protocols[13].

1.7 Objectives of the study

1. To review the cryptographic properties of Boolean functions.
2. To extend the properties analogous to Boolean functions to functions over Z_q.
3. To derive the methods for extracting and formulating function over Z_q with specific cryptographic properties.
4. To Design and analyse non-binary cryptographic primitives. .

1.8 Relevance of the study

New approaches to data protection are necessary as technology develops. Thus, it is necessary that the current cryptographic primitives to be improved at any cost. The development of quantum computation gives way to cryptographic primitives based on non-binary (many-valued) logic. Also, an adversary targeting a cryptosystem is not limited by the representation of the scheme. The existing cryptographic schemes which is based on binary system also be represented using non-binary primitives. Thus, there is a scope for developing cryptanalytical methods based on non-binary primitives. In order to achieve this aim, one may have a deep knowledge of cryptographic primitives on non-binary logic. Functions over the rings Z_q are one of these primitives. Therefore, it is essential to analyse the cryptographic characteristics of those functions to design cryptographic schemes based on them.

CHAPTER 2

Review of Literature and Preliminaries

2.1 Review of Literature

Boolean functions are the building blocks of most of the well-known block ciphers like AES, DES, etc. So Boolean functions have an important role in the design of cryptographic algorithms based on binary logic. There are several articles analysing the cryptographic characteristics of Boolean functions. When it comes to ensuring the security of a cryptographic construction against various forms of cryptanalysis, Boolean functions employed in cryptographic engineering should meet specific requirements. High algebraic degree, balancedness, correlation immunity, resiliency, nonlinearity, and algebraic immunity are some of the cryptographic characteristics of the Boolean functions. Analogous properties of functions over Z_q (q-functions are comparat) ively less studied.

It is noted that the synthesis of non-binary cryptographic constructions is receiving more focus at the current stage in the development of cryptographic algorithms[14]. The primary aim is to create new and enhance

existing techniques for the synthesis of cryptographically stronger primitives based on q-functions, which are subsequently utilised to build such cryptographic algorithms. To develop such techniques, one must have deep knowledge of the cryptographic characteristics of q- functions.

In the paper [15] the cryptographic properties such as propagation criterion and Strict Avalanche Criterion(SAC) of Boolean function are discussed. The concept of propagation criterion extended to functions of q-valued logic. The definition of propagation criterion was extended to ternary functions in [16]. In this paper, the authors synthesised the ternary functions of length 9 of two variables that possess strict avalanche characteristics. The author of the article [17] elaborates on the avalanche characteristics of quaternary functions and synthesized 7680 balanced quaternary functions with strict avalanche criteria. These functions offer better performance if utilised for cryptographic construction based on quaternary logic. The avalanche characteristics of S-boxes constructed based on Nyberg design were studied in line with functions over Z_q in [18]. These S-boxes can be described by 4-functions and 16-functions as their components. A detailed study on the avalanche characteristics of the S-boxes constructed using various irreducible polynomials is carried out in terms of Boolean functions, 4-functions, and 16-functions. It was pointed out that the S-box constructed with the polynomial $x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$ possesses the smallest deviation from the expected value required to satisfy strict avalanche criterion in binary component representation, whereas the polynomials $x^8 + x^7 + x^6 + x + 1$ and $x^8 + x^5 + x^4 + x^3 + x^2 + x + 1$ give best result in terms of strict avalanche criterion when the S-box is represented by component 4-functions and 16-functions respectively.

Nonlinearity is another important cryptographic characteristic. The nonlinearity of a Boolean function can be computed on the basis of the

Walsh transform[19] with the help of the Hadamard matrix. The nonlinearity of q-functions can be computed with the help of Vilenkin-Chrestenson transform[20]. The nonlinear properties of the quaternary functions of lengths 4 and 16 were analyzed in [21] and classified the quaternary functions into different spectral classes based on the nonlinearity values. The article [22] analyses the nonlinearity properties of the Rijindael like S –boxes when representing the components using 4-functions and 16-functions with various irreducible polynomials and found the optimal irreducible polynomials on F_2. It was found that the S-box constructed on the basis of the polynomial $x^8 + x^7 + x^5 + x + 1$ has the highest nonlinearity values when presented with the components as 4-functions and the polynomial $x^8 + x^7 + x^3 + x + 1$ gives the optimal nonlinearity in 16-valued component representation. The nonlinearity of AES S-box is analyzed in terms of 4-functions and 16-functions in [23] and the study shows that the four components 4-functions of the S-box have the nonlinearity 219.5034, 216.9412, 216.5538 and 219.284 and hence the S-box has the nonlinearity 216.9412 when presented as component 4-functions. When the S-box is presented in terms of component 16-functions, there are two components and these components possess nonlinearity values 213.8184 and 212.4385 and hence the S-box has nonlinearity 212.4385.

Rothas [24] introduced bent functions, a class of functions with cryptographic importance. A handful of research articles are available on Boolean bent functions, their construction, and classifications.[25, 26]. Bent-sequences, which have uniform absolute values for spectral coefficients and the highest level of nonlinearity take a unique place among the more advanced algebraic structures for modern science and technology. Due to their constant spectral characteristics, bent functions are used as C-codes in MC-CDMA (Multi Code Code Division Multiple Access)

technologies to lower the PAPR (Peak-to-Average Power Ratio) of signals [27], also in cryptography to produce extremely nonlinear S-boxes [28] and for developing pseudo-random key sequence generators [29, 30]. The use of many-valued logic concepts is another advancement in the field of communications and information security that is becoming more common. Technological advancement demands new methods for cryptographers, so the algebraic constructions over ternary and quaternary logic emerge and develop. The method for the generation of ternary pseudo-random key sequence generation was discussed in [31]. The article [32] provides a method for synthesising ternary bent sequences in two variables and lists all 486 ternary bent sequences of length 9. The research on ternary bent sequences of three variables was conducted in [33]. An algorithm on the basis of the Reed-Muller transformants domain that reduces the time complexity by 6561 times less than exhaustive search was developed and showed that the number of ternary bent sequences of length 27 is 12623364 and it is classified into 6 classes according to their weight structures.

The correlation attack was coined by Siegethaler in [34] and the original attack was given in[35]. The functions which have resistance against correlation attacks are known as correlation immune functions. The Correlation immunity of order m (m-CI) was studied in [34]. The cryptographic characteristics such as correlation immunity and resiliency of functions on binary logic were extensively studied in [34, 35]. The extension of the concept to functions of q-valued logic, where q is a prime number was introduced in[36] and presented three characterisations, Matrix, Fourier transform, and orthogonal array characterisations of correlation immunity of functions over the Galois field of prime characteristic. This is further extended to rings over Z_q in[37]. A detailed study of correlation immunity of ternary functions and a method to synthesise the complete class of

correlation immune ternary functions in two variables was given in[38] and the complete class of correlation immune functions of length 9 were synthesised and found the balanced correlation immune functions which can be the basis for constructing quality S-boxes.

The article [39] discusses a method for the construction of S-boxes with correlation immune characteristics both in a Boolean and quaternary sense. Using the method they were able to synthesise 18304 S-boxes of length 16 satisfying the criterion of correlation immunity in both, binary and quaternary sense.

The correlation immunity of the functions over Z_q is further investigated in [40] and conducted an analysis of the correlation immunity of the component functions of the S-boxes of the AES and Kalyna, a block cipher developed by Ukraine. They analysed the correlation immunity of the component functions of these S-boxes by expressing them in terms of Boolean, quaternary, and 16- functions. They found that the components of S-boxes of both block ciphers do not satisfy correlation immunity in binary, quaternary, and 16-valued sense. To quantify if the S-box complies with the correlation immunity requirement, they introduced maximum and integral deviation of the component q- functions, and the S-box, from the compliance with the criterion that the output be independent of its input variables. These two indicators provide a method for a comparative study of the correlation properties of the cryptographic constructions on many-valued logic. The studies [40] show that the indicators of the S-box of both the block cipher AES and Kalyna when expressed in terms of q-functions have a tendency to grow up with the increase of the value of q.

To measure the input-output correlation of the component q-functions, the input-output correlation coefficient and the correlation matrix consisting of the absolute values of the correlation coefficient were introduced in[41]

In 1999, Pieprzyk and Qu [42] proposed a sub-class of Boolean functions named Rotation symmetric Boolean functions(RSBF). The article appeared in 2002 [43] presents a formula to find the number of rotation symmetric functions in n variables using Burnside's lemma. They also found the complete class of rotation symmetric bent functions of an even number of variables up to 8 using a computer search. The paper published in 2003 [44] discusses some significant findings of the homogeneous rotation symmetric Boolean functions and on rotation symmetric cycles. The formula to find the number of rotation symmetric non-binary functions over the Galois field of characteristic p is presented in [45]. In the article published in 2004 [46] the authors present an effective method for extraction of rotation symmetric Boolean bent and correlation immune functions with reduced computational complexity. The article[48] presents a theoretical method for the construction of rotation symmetric Boolean functions with maximum algebraic immunity and they found rotation symmetric functions of 7,9 and 11 variables with maximum algebraic immunity and having nonlinearities 56,240, and 984 respectively.

The extension of rotation symmetry to functions over Z_q and the intriguing characteristics of such q-functions received much discussion from authors in[49]. An article [50] enunciates the bentness of the ternary symmetric functions.

The fundamental element of symmetric key algorithms in cryptography that executes substitution is called an S-box (substitution-box). They usually serve to ensure Shannon's property of confusion in block ciphers by masking the connection between the key and the cipher text. The construction of binary S- boxes over the Galois field are covered in several articles [51, 52]. The design of S-boxes that satisfy specific cryptographic properties was discussed in [53, 54]. The construction is extended to

include non-binary cases and constructed S-boxes with zero input-output correlation[54]. A block symmetric algorithm based on non-binary logic is designed in[55]. The non-binary S-box was utilised in this algorithm and this block symmetric algorithm was successfully applied to image encryption. The authors in[56] constructed S-boxes over the Galois field of characteristics 3 and 5 and its isomorphic forms and analysed its nonlinear properties.

The block cipher AES is generalized to the Galois field of any characteristic in the article[57]. They describe a generalisation of AES that uses ternary logic, and the same algorithm used to encrypt digital images.

2.2 Preliminaries

2.2.1 Galois Field

A field is a commutative ring with unity and without zero divisors. Associated with each prime number p, there is a field of order p which is denoted by F_p. The characteristic of a field is the least positive integer m such that $ma = 0$ for every element a in the field and it must be a prime number[58]. A finite field is a field with a finite number of elements. The set of non-zero elements in the field forms a cyclic group under multiplication, which is denoted by ${F_p}^*$.

If F_p is a finite field, then the set of all polynomials in the variable x with coefficients from F_p forms the polynomial ring denoted by $F_p[x]$. Let $p(x) \in F_p[x]$ be a polynomial of degree n, then the equivalence class of all polynomials congruent to $p(x)$ forms a quotient ring denoted by $\frac{F_p[x]}{(p(x))}$. This quotient ring can be identified as a vector space of all the polynomials with a degree less than n under the operations, usual polynomial addition and multiplication modulo $p(x)$.

The polynomial $p(x) \in F_p[x]$ is said to be irreducible if it has no roots

in F_p or in other words $p(x)$ is irreducible if it cannot be factored into polynomials in $F_p[x]$ of degree less than $p(x)$. Monic polynomials are those with a leading coefficient 1. A polynomial that generates all the members in an extension field from a base field is known as a primitive polynomial. Moreover, primitive polynomials are irreducible polynomials.

The number of polynomials that are irreducible and the number of primitive polynomials over F_p of degree n can be determined using the equations provided in [59]. The formula to determine the number of polynomials that are irreducible is as follows:

$$\frac{1}{n}\sum_{d/n}\mu(d)p^{\frac{n}{d}} \tag{2.2.1}$$

Where $\mu(d)$ is the Mobius function. The formula to find the number of primitive polynomials is;

$$\frac{1}{n}\phi(p^n-1) \tag{2.2.2}$$

Where $\phi(d)$ is the Euler-Totient function.

If $p(x)$ is a polynomial irreducible over F_p of degree n, the quotient ring $\frac{F_p[x]}{\langle p(x)\rangle}$ forms a field with p^n elements known as Galois field denoted by $GF(p^n)$. We call $GF(p^n)$ the extension field of F_p, a field containing F_p. If $p(x)$ is irreducible then $kp(x)$ is also irreducible, where k is a scalar. So it is enough to consider the monic irreducible polynomials for field construction.

The method of construction and existenc of the Galois field for a given prime number p and for any positive integer n is addressed in the theorem given in[58].

Theorem 2.2.1. *[58] For a prime p and monic irreducible polynomial p(x) in $F_p[x]$ of degree n, the ring $F_p[x]/(p(x))$ is a finite field of order p^n.*

The Galois field $GF(p^n)$ can also be viewed as a vector space over F_p with dimension n. Since $GF(p^n)$ contains polynomials of degree at most $n-1$, the set $\{1, x, x^2, ..., x^{n-1}\}$ forms a basis. Therefore, $GF(p^n)=\{a_{n-1}x^{n-1}+ a_{n-2}x^{n-2} + ... + a_2x^2 + a_1x + a_0, \quad a_i \in F_p, i = 0, 1, 2...n\}$. The members of the field $GF(p^n)$ can also be written in short form as a string of elements in F_p as $GF(p^n) = \{a_{n-1}a_{n-2}...a_2a_1a_0, \quad a_i \in F_p, i = 0, 1, 2...n-1\}$.

For example, consider the field $GF(3^2)$ constructed using the monic irreducible and primitive polynomial $f(x) = x^2 + 2x + 2$ of degree 2 over F_2. Since $f(x)$ is a primitive polynomial, all the elements of $GF(3^2)$ can written as the power of the root of $f(x)$. Let β be a root of $f(x)$. There fore, $f(\beta) = \beta^2 + 2\beta + 2 = 0$. Since addition is performed modulo 3, it follows that $\beta^2 = -2\beta - 2 = \beta + 1$, $\beta^3 = 2\beta + 1$, $\beta^4 = 2$, $\beta^5 = 2\beta$, $\beta^6 = 2\beta + 2$, $\beta^7 = \beta + 2$, $\beta^8 = 1$. Thus, it follows that the Galois field, $GF(3^2) = \{0, 1, \beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6, \beta^7\} = \{0, 1, \beta, \beta+1, 2\beta+1, 2, 2\beta, 2\beta+ 2, \beta + 2\}$. This can also be described by a set of strings of ternary numbers (members of F_3) as $GF(3^2) = \{00, 01, 10, 11, 21, 02, 20, 22, 12\}$.

Modern block ciphers make use of constructions based on the Galois field for better performance. Additionally, constructions on many-valued logic utilise the advantageous characteristics of the Galois field.

2.2.2 Cryptographic Properties of Boolean Functions

Let $F_2 = \{0, 1\}$ be a field with operations addition modulo 2 and multiplication modulo 2. ${F_2}^n$, the n copies of F_2 forms a vector space over the base field F_2.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in {F_2}^n$, then $\mathbf{x}$ can be identified as an integer,

$$(\sum_{1}^{n} x_{n-i-1}2^i) mod 2^n \tag{2.2.3}$$

For example, the element $(1, 0, 1, 1) \in {F_2}^4$ corresponds to the integer 11.

This representation gives an ordering of the elements in ${F_2}^n$ with respect to the corresponding integer values, known as lexicographic ordering. The elements are ordered in increasing order of the corresponding integer values.

The weight of a binary vector is the number of 1's in it. Associated with each vector **x**, there is a set denoted by Sup(**x**) containing all non-zero position values in **x** known as support of **x**. That is Sup(**x**) = $\{k : x_k \neq 0\}$. Let **x**, **y** $\in {F_2}^n$, then d(**x**,**y**) denote the distance between **x** and **y**, the number of positions in which they differ.

A Boolean function in n variables is an arbitrary function whose domain is ${F_2}^n$ and whose range is F_2 and it is named in honor of the British mathematician and philosopher George Boole (1815-1864). The cardinality of the set of all Boolean functions is equal to 2^{2^n}.

Let $f : {F_2}^n \rightarrow F_2$ be a Boolean function in n variables, then it can be described in various ways. The truth table (TT), which depicts the function values for each of the possible values of its input variables, is one mode of representation. In this mode, the function f can be depicted as $f = [f(0, 0, 0, ..., 0), f(0, 0, 0, .., 1), ..., f(1, 1, 1, ..., 1)]^T$.The integer expression of these input vectors according to 2.2.3 renders the function expression $f = [f(1), f(2), ..., f(2^n - 1)]^T$. Corresponding to each Boolean function, there is a function $(-1)^f = 1 - 2f$ that assumes values of the set {-1,1}. The vectorial representation of this function is $(-1)^f = (-1)^{f(1)}, (-1)^{f(2)}, ..., (-1)^{f(2^n-1)}]^T$ and is known as Polarity truth table (PTT). The transpose of the TT and PTT forms $2^n \times 1$ matrices denoted by $[f]$ and $[(-1)^f]$ respectively.

A Boolean function can also be represented as a polynomial in the quotient ring $F_2[x_1, x_2, ..., x_{n-1}]/(x_1^2 - 1, x_2^2 - 1, ..., x_n 2 - 1)$ known as algebraic normal form (ANF). Being a linear transformation, ANF has a matrix representation.

Let A_f denote $2^n \times 1$ column matrix containing the coefficients of the polynomial in the ANF of the Boolean function f, then

$$A_f = A_n.f(mod 2) \tag{2.2.4}$$

Where A_n is the matrix produced by the recurrence relation;

$$A_{n+1} = \begin{bmatrix} A_n & 0 \\ A_n & A_n \end{bmatrix}, \text{ where } A_0 = 1$$

It is to be noted that ${A_n}^2 = I_{2^n}$, identity matrix of order 2^n and hence the truth table of the function f can be obtained by the following expression;

$$f = A_n.A_f \tag{2.2.5}$$

The algebraic degree is the degree of the longest term whose coefficient is non-zero.

Walsh transform can be utilised for the various analysis of a Boolean function. Let $f : {F_2}^n \to F_2$ be a Boolean function, the Walsh transform of f at a vector $u \in {F_2}^n$ is uniquely defined as[19];

$$W_f(u) = \sum_{x \in {F_2}^n} (-1)^{f(x)+\langle x,u \rangle} \tag{2.2.6}$$

Where the addition is performed the modulo 2.

Walsh spectrum is also computed in terms of matrix multiplication. Let W_f denote $2^n \times 1$ column matrix that represents the Walsh transform values of the function f for all possible values of $u \in {F_2}^n$ arranged lexicographical

order of u, we call it by Walsh spectrum. Then the Walsh spectrum of f is obtained as;

$$W_f = H_n.[(-1)^f] \tag{2.2.7}$$

Where the matrix H_n is known as the Hadamard matrix and is obtained using the recursive relation;

$$\mathrm{H}_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}, \mathrm{H}_0 = 1.$$

Research regarding the construction of Boolean functions with strong cryptographic properties is an unending area. In order to enhance the cryptographic strength of the ciphers constructed on the basis of Boolean functions, the component Boolean functions meet some desirable cryptographic properties.

Nonlinearity is one among the desirable characteristics of a BF. For the construction of secure systems, highly nonlinear Boolean functions are preferred. Walsh spectrum can be utilized to measure nonlinearity, which is the distance of the function from the set of all linear and affine functions. That is the nonlinearity N_f of the Boolean function f is;

$$N_f = \min\{d(f, g) : g \text{ is an affine function}\} \tag{2.2.8}$$

For a BF f, the nonlinearity N_f is measured from the Walsh spectrum as follows.

$$N_f = 2^{n-1} - \frac{1}{2} Max_{u \in F_2{}^n} |W_f(u)| \tag{2.2.9}$$

The nonlinearity of a BF is upper bounded by $2^{n-1} - 2^{\frac{n}{2}-1}$.

Bent functions are the functions that have the worst possible approximation by a linear or affine function. Stated otherwise, a function is bent if it is a maximally nonlinear functions. Additionally, bent functions are

described in terms of the Walsh transform. A Boolean function f is bent if it has constant absolute spectral values. That is $|W_f(u)| = 2^{\frac{n}{2}}$ for every values u in its domain, so that f attains the maximum nonlinearity.

The propagation criterion or avalanche criterion is another important characteristic of the BFs that is directly connected with diffusion, introduced by Shannon. The potential of a cryptographic design to transmit minute changes in the input text or key element to the full ciphertext is determined by the propagation criterion[60]. There are two different types of error propagation: propagation in the direction of a specific vector u and propagation criterion of a specific order k, which is measured along a vector of weight k. Both the propagation criteria are interrelated to compute the propagation characteristics of a cipher.

The strict avalanche criterion (SAC) is the propagation criterion in the strict sense and it was introduced by Webster and Tavares[15]. The SAC is the propagation criterion of order one. If a single bit change in the input results in a 50 percent chance of changing the output bit, the Boolean function is said to satisfy SAC.

The propagation criterion of Boolean functions is quantified by means of the derivative as a mathematical tool[61]. The bias of the output probability distribution of the derivative in the direction of a vector of weight k determines the avalanche criterion of order k. The SAC is the measure of the bias of the derivatives of the Boolean function in the direction of vectors of weight 1.

2.2.3 Functions over Rings Z_q (Function of q-valued Logic)

Consider the ring $Z_q = \{0, 1, 2. \ldots, q-1\}$ with operation addition modulo q and multiplication modulo q. If q is a prime number, then Z_q turns into a field. ${Z_q}^n$ is the n copies of Z_q, in this thesis, we denote it by $V(n, q)$. ${Z_q}^n$ can also be considered as a vector space of dimension n over Z_q, and this

vector space is used in the generation of functions of q-valued logic.

Definition 2.2.2. (Function over Z_q). A function over Z_q(q-function or function of q-valued logic) in n variables is an arbitrary function whose domain is $V(n, q)$ and range is Z_q.

The weight of a string of members of Z_q is defined as the count of non-zero elements in it. Thus, the weight of a q-function is the number of non-zero elements in it.

2.2.4 Representation of Functions over Z_q

Functional representation is an important aspect of analysing various characteristics of that function. This section briefly discusses various means of representations of functions of over Z_q.

2.2.5 Truth table

Analogous to Boolean functions, there are various means to represent a q-function. One way of representation is by employing truth tables. Truth table representation tabulates all possible combinations of input values and their respective outputs ordered lexicographically or simply represents the output values as a string of members of the underlying set. Let f be a q-function in n variables, then its truth table can be represented as a string of length q^n as $f = [f(0, 0, \ldots 0)f(0, 0, \ldots, 1)...f(q-1, q-1, ..., q-1)]^T$.

The table 2.1 represents a function of 16 points in different domains of variables. When expressed as a truth table, a function of length 16 can be written as a function of 4 variables in the binary domain but a function of only 2 variables in the quaternary domain. This is one benefit of q-valued representations so that more data can be embedded with less storage. The 4-valued function in table 2.1 when represented as an array or string of quaternary numbers looks like $[0123210312033102]^T$.

Table 2.1: *Truth table of functions of length 16*

$x_1x_2x_3x_4$	$f(x_1x_2x_3x_4)$	y_1y_2	$f(y_1y_2)$
0000	f(0000) = 1	00	f(00) = 0
0001	f(0001) = 0	01	f(01) = 1
0010	f(0010) = 1	02	f(02) = 2
0011	f(0011) = 1	03	f(03) = 3
0100	f(0100) =0	10	f(10) = 2
0101	f(0101) = 1	11	f(11) = 1
0110	f(0110) = 0	12	f(12) = 0
0111	f(1000) = 1	13	f(13) = 3
1000	f(1000) = 1	20	f(20) = 1
1001	f(1001) = 1	21	f(21) = 2
1010	f(1010) = 1	22	f(22) = 0
1011	f(1011) = 0	23	f(23) = 3
1100	f(1101) = 1	30	f(30) = 3
1101	f(1101) = 1	31	f(31) = 1
1110	f(1110) = 1	32	f(32) = 0
1111	f(1111) = 1	33	f(33) = 2

2.2.6 Complex Exponent Form

Associated with each q-function f= $[f_0 f_1 ... f_{q^{n-1}}]^T$, there is a function known as the *complex exponent form*, defined as $F = [\omega^{f_0}\omega^{f_1}...\omega^{f_{q^{n-1}}}]^T$ where ω is the q^{th} root of unity. This representation is also known as the *polarity truth table*. The complex exponent form of the function is extremely helpful in analysing the spectral properties of functions over Z_q. The complex exponent form of the 4- valued (quaternary) function in table 2.1 is $F = [1, i, i^2, i^3, i^2, i, 1, i^3, i, i^2, 1, i^3, i^3, i, 1, i^2]^T$, where i is the fourth root of unity.

2.2.7 Algebraic Normal Form

A multivariable polynomial can be used to describe a function over Z_q in a unique way. This representation of a function as a polynomial is known as algebraic normal form.

Definition 2.2.3 (Algebraic normal form (ANF)). *[62] An algebraic normal form of a q- function is a polynomial f over Z_q of $deg(f) \leq q-1$ with coefficients $a_i \in \{0, 1, ..., q-1\}$, containing the operations "Sum in the Galois field $GF(q)$" and "Multiplication in the Galois field $GF(q)$".*

It is to be noted that if q is a prime number then the operations are reduced to addition modulo q and multiplication modulo q. The ANF can be computed in the Reed-Muller Transformants domain[5].

Let $f : V(n, q) \longrightarrow Z_q$ be a function of q-valued logic, the algebraic normal form of the function f is given by;

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{q^n-1} a_i {x_1}^{i_1} ... {x_n}^{i_n}. \tag{2.2.10}$$

Where $a_i \in Z_q$, $i = 1, 2, ..., q^n - 1$ are the ANF coefficients and $i_j = 1, 2, ..., q-1$.

Let A_f be the vector of ANF coefficients then this coefficient vector can be computed using the equation 2.2.11 [5].

$$A_f = f.R_{q^n} \tag{2.2.11}$$

Where R_q^n is the Reed-Muller transform matrix.

The inverse $R_{q^n}^{-1}$ of the Reed-Muller transform matrix described by the Kronecker product [5].

$$R_{q^n}^{-1} = \bigotimes R_q^{-1} \tag{2.2.12}$$

Definition 2.2.4 (Algebraic degree). *The algebraic degree is the highest sum of degrees of the monomial in the algebraic normal form whose coefficient is non-zero.*

A function is considered homogeneous if each monomial in the ANF is of the same degree.

For ternary function(q =3), the Reed-Muller transform matrix can also be specified by the following recurrence relation[5].

$$R_{3^n}^{-1} = \begin{bmatrix} R_{3^{n-1}} & 0_{3^{n-1}} & 0_{3^{n-1}} \\ R_{3^{n-1}} & R_{3^{n-1}} & R_{3^{n-1}} \\ R_{3^{n-1}} & 2R_{3^{n-1}} & R_{3^{n-1}} \end{bmatrix}, R_3^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

The matrix $0_{3^{n-1}}$ is the square matrix of order 3^{n-1} with all its entries 0. The inverse of $R_{3^n}^{-1}$ in the field $GF(3)$ is the matrix of ANF transformation.

For a ternary function f in two variables x_1 and x_2 and with the vector of ANF coefficients, $A_f = [a_{00}\ a_{01}\ a_{02}\ a_{10}\ a_{11}\ a_{12}\ a_{20}\ a_{21}\ a_{22}]$.The general structure of ANF of f is $f(x_1, x_2) = a_{00} + a_{01}x_2 + a_{02}{x_2}^2 + a_{10}x_1 + a_{11}x_1x_2 + a_{12}x_1{x_2}^2 + a_{20}{x_1}^2 + a_{21}{x_1}^2x_2 + a_{22}{x_1}^2{x_2}^2$.

For example, consider a ternary function f in two variables whose truth table representation is $f = [0\ 0\ 0\ 1\ 0\ 2\ 2\ 0\ 1]^T$. The Reed-Muller transform matrix of order 9x9 for computing the Reed-Muller transform coefficients is

$$R_9 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Then the vector of ANF Coefficients obtained by the multiplication of f with R_9 is $A_f = [0\ 0\ 0\ 1\ 2\ 0\ 0\ 0\ 0]$. Thus, the algebraic normal form is $f(x_1, x_2) = x_1 + 2x_1x_2$ and, the algebraic degree of f is equal to 2.

Appendix A and Appendix B describe algorithms for finding the ANF and algebraic degree of ternary functions of length 9 and 27 respectively, which are used for computation in this thesis.

The algebraic normal form of a quaternary function is described over the Galois field $GF(4)$ constructed using the unique polynomial $x^2 + x + 1$ irreducible over F_2. This unique monic irreducible polynomial determines the multiplication. Addition is performed bit-wise modulo 2. Addition and multiplication in the field $GF(4)$ are given in the table 2.2.

Table 2.2: *Addition and Multiplication in GF(4)*

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

.	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

When $q = 4$, the inverse of the Reed-Muller transform matrix is given by the recurrence relation according to[5];

$$R_{4^n}^{-1} = \begin{bmatrix} R_{4^{n-1}} & 0_{4^{n-1}} & 0_{4^{n-1}} & 0_{4^{n-1}} \\ R_{4^{n-1}} & R_{4^{n-1}} & R_{4^{n-1}} & R_{4^{n-1}} \\ R_{4^{n-1}} & 2R_{4^{n-1}} & 3R_{4^{n-1}} & R_{4^{n-1}} \\ R_{4^{n-1}} & 3R_{4^{n-1}} & 2R_{4^{n-1}} & R_{4^{n-1}} \end{bmatrix}, R_4^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 1 \end{bmatrix}$$

The general expression for the algebraic normal form of a quaternary function in two variables is $f(x_1, x_2) = a_{00} + a_{01}x_2 + a_{02}{x_2}^2 + a_{03}{x_2}^3 + a_{10}x_1 + a_{11}x_1x_2 + a_{12}x_1{x_2}^2 + a_{13}x_1{x_2}^3 + a_{20}{x_1}^2 + a_{21}{x_1}^2x_2 + a_{22}{x_1}^2{x_2}^2 + a_{22}{x_1}^2{x_2}^3 + a_{30}{x_1}^3 + a_{31}{x_1}^3x_2 + a_{32}{x_1}^3{x_2}^2 + a_{33}{x_1}^3{x_2}^3$.

For example, consider a quaternary function f in two variables whose truth table representation is given by $f = [0000013203210213]^T$. The Reed-Muller transform matrix of order 16x16 to be used for the computation is given by;

$$R_{16} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 & 0 & 3 & 2 & 1 & 0 & 2 & 1 & 3 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 3 & 1 & 2 & 0 & 2 & 3 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 3 & 2 & 0 & 2 & 1 & 3 & 0 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 2 & 3 & 1 & 0 & 3 & 1 & 2 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 3 & 2 & 0 & 1 & 3 & 2 & 0 & 1 & 3 & 2 & 0 & 1 & 3 & 2 \\
0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Then $A_f = [0, 0, 0, 0, 0, 1, 3, 0, 0, 3, 2, 0, 0, 0, 0, 2]$. Thus its algebraic normal form is $f(x_1, x_2) = x_1x_2 + 3x_1{x_2}^2 + 3{x_1}^2x_2 + 2{x_1}^2{x_2}^2 + 2{x_1}^3{x_2}^3$ and hence the algebraic degree of the function f is 6.

For the computation of ANF and the algebraic degree of the quaternary functions, Appendix C provides an algorithm in the Python programming language.

2.2.8 Nonlinearity of Functions over Z_q

The coefficient of Vilenkin -Chrestenson transform [20] can be used to determine the nonlinearity of functions over Z_q. The Vilenkin-Chrestenson coefficients of a q- function f for $u \in V(n,q)$ can be computed using expression 2.2.13.

$$.\Phi_f(u) = \sum_{x \in V(n,q)} F(x)\omega^{\overline{\langle u,x \rangle}} \tag{2.2.13}$$

Where $F(x)$ is the complex exponent form of the function $f(x)$ and $w = e^{\frac{2\pi i}{q}}$ is the q^{th} root of unity. We call the spectral coefficient as the *spectral value*. The vector consists of all the spectral values of a q-function at each value in its domain, which we call, the *Vilenkin-Chrestenson spectrum* of the q-function. The Vilenkin -Chrestenson spectrum of a q- function can be determined by employing the Vilenkin-Chrestenson transform matrix of order q^n. For $q = 3$, the following recurrence relation can be used to generate this matrix [63].

$$V_{3^{k+1}} = \begin{bmatrix} V_{3^k} & V_{3^k} & V_{3^k} \\ V_{3^k} & V_{3^k} + \mathbf{1} & V_{3^k} + \mathbf{2} \\ V_{3^k} & V_{3^k} + \mathbf{2} & V_{3^k} + \mathbf{1} \end{bmatrix}, \text{ where } V_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

Here **k** is the square matrix of order 3^k with all entries equal to k and the matrix addition is performed modulo 3.

For $q = 4$, the Vilenkin-Chrestenson transform Matrix is obtained from the following recurrence relation[21].

$$V_{4^{k+1}} = \begin{bmatrix} V_{4^k} & V_{4^k} & V_{4^k} & V_{4^k} \\ V_{4^k} & V_{4^k} + \mathbf{1} & V_{4^k} + \mathbf{2} & V_{4^k} + \mathbf{3} \\ V_{4^k} & V_{4^k} + \mathbf{2} & V_{4^k} & V_{4^k} + \mathbf{2} \\ V_{4^k} & V_{4^k} + \mathbf{3} & V_{4^k} + \mathbf{2} & V_{4^k} + \mathbf{1} \end{bmatrix}, \text{ where } V_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 0 & 2 \\ 0 & 3 & 2 & 1 \end{bmatrix}$$

The spectral coefficients of a q- function in n variables are the values obtained by multiplying the function with the conjugate of the Vilenkin-Chrestenson matrix of order q^n by expressing both in its complex exponent form[**?**]. This conversion to complex exponent form can be done using the transformation $\{0, 1, 2, ..., q-1\} \rightarrow \{e^{\frac{2\pi i}{q}0}, e^{\frac{2\pi i}{q}1}, e^{\frac{2\pi i}{q}2}, ..., e^{\frac{2\pi i}{q}(q-1)}\}$.

$$\Phi_f(u) = f.\overline{V_{q^n}} \tag{2.2.14}$$

Where $\overline{x}$ represents the complex conjugate of x. The nonlinearity of a q-function is [63];

$$N_f = \begin{cases} q^{n-1} - \frac{1}{2}Max_{u \in V(n,q)}|W_f(u)| & if \ \ q = 2 \\ q^n - Max_{u \in V(n,q)}|\Phi_f(u)| & if \ \ q \geq 3 \end{cases} \tag{2.2.15}$$

For the computation of the nonlinearity of the ternary function of length 9, one may require the matrix of the Vilenkin-Chrestenson transform of order 9. This matrix after performing the unique complex exponential transformation $\{0, 1, 2\} \rightarrow \{1, e^{\frac{2\pi}{3}i}, e^{\frac{4\pi}{3}i}\}$is;

$$V_9 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} \\ 1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} \\ 1 & 1 & 1 & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} \\ 1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 & e^{\frac{4\pi}{3}i} & 1 & e^{\frac{2\pi}{3}i} \\ 1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 \\ 1 & 1 & 1 & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} \\ 1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 \\ 1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 \end{bmatrix}$$

For example, consider a ternary function f in two variables whose truth table is $f = [012221011]^T$. Then its complex exponent form by applying the transformation is $F = [1\ e^{\frac{2\pi}{3}i}\ e^{\frac{4\pi}{3}i}\ e^{\frac{4\pi}{3}i}\ e^{\frac{2\pi}{3}i}\ 1\ e^{\frac{2\pi}{3}i}\ e^{\frac{2\pi}{3}i}]^T$. Then the vector of the Vilenkin-Chrestenson spectrum is $\Phi_f = [e^{\frac{4\pi}{3}i} + 2e^{\frac{2\pi}{3}i}\ 4 + 2e^{\frac{4\pi}{3}i}\ 4 + 2e^{\frac{4\pi}{3}i}\ 2e^{\frac{2\pi}{3}i} + e^{\frac{4\pi}{3}i}\ 5 + 4e^{\frac{2\pi}{3}i}\ e^{\frac{4\pi}{3}i} + 2e^{\frac{4\pi}{3}i}\ 4 + 2e^{\frac{4\pi}{3}i}\ 4 + 2^{\frac{4\pi}{3}i}\ e^{\frac{4\pi}{3}i} + 2e^{\frac{4\pi}{3}i}]$.

The maximum modulus of the spectral values is equal to 4.5826; hence, the nonlinearity is equal to $3^2 - 4.5826 = 4.4174$.

Appendix D and Appendix E provide algorithms that compute the nonlinearity value of the ternary and quaternary functions respectively, for a given number of variables.

A q-function is a bent function if the spectral values are flat and have the highest possible nonlinearity. In other words, it has a uniform absolute value for every Vilenkin-Chrestenson spectral coefficient for all potential values in its domain. This happens if the modulus of each of the spectral coefficients is equal to $q^{\frac{n}{2}}$. A bent q-function in n variables attain the maximum nonlinearity $q^n - q^{\frac{n}{2}}$.

For example, Consider a ternary function f in 2 variables whose truth table is $f = [000012021]^T$. The vector of Vilenkin-Chrestenson spectral coefficients is $[3\ 3\ 3\ 3\ 3\ e^{\frac{4\pi}{3}i}\ 3e^{\frac{2\pi}{3}i}\ 3e^{\frac{2\pi}{3}i}\ 3e^{\frac{2\pi}{3}i}\ 3e^{\frac{4\pi}{3}i}]$. The absolute value of the spectral coefficients is uniform and the nonlinearity NL =6, the highest possible nonlinearity of a function of this kind. Therefore f is a bent function.

Now consider a quaternary function f whose truth table is $f = [03311323\ 33230113]^T$ and the polarity truth table is F= [1 -i -i -i i i -i -1 -i -i -i -1 -i 1 i i -i].

The matrix of the Vilenkin-Chrestenson transformation of order 16 is

$$_{16} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & i & -1 & -i & 1 & i & -1 & -i & 1 & i & -1 & -i & 1 & i & -1 & -i \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & 1 & 1 & i & i & i & i & -1 & -1 & -1 & -1 & -i & -i & -i & -i \\
1 & i & -1 & -i & i & -1 & -i & 1 & -1 & -i & 1 & i & -i & 1 & i & -1 \\
1 & -1 & 1 & -1 & i & -i & i & -i & -1 & 1 & -1 & 1 & -i & i & -i & i \\
1 & -1 & 1 & -1 & i & -1 & -i & 1 & -1 & i & 1 & -i & -i & -1 & i & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & i & -1 & -i & 1 & -1 & -i & 1 & 1 & i & -1 & -i & 1 & -1 & -i & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & i & 1 & 1 & -1 & 1 & -1 & 1 & -1 & i & 1 \\
1 & 1 & 1 & 1 & -i & -i & -i & -i & -1 & -1 & -1 & -1 & i & i & i & i \\
1 & i & -1 & -i & -i & 1 & i & -1 & -1 & -i & 1 & i & i & -1 & -i & 1 \\
1 & -1 & 1 & -1 & -i & i & -i & i & -1 & 1 & -1 & 1 & i & -i & i & -i \\
1 & -1 & 1 & -1 & -i & -1 & i & 1 & -1 & i & 1 & -i & i & -1 & -i & 1
\end{bmatrix}$$

Then $\Phi_f = [-4i, 4, 4i, 4, 4i, 4i, 4, 4, -4i, -4, -4i, 4, 4, -4, -4i, 4i]$. Since the absolute value of the spectral coefficients is uniform and is equal to $4^{\frac{2}{2}} = 4$, it is obvious that f is a quaternary bent function. The nonlinearity of f is equal to 12, which is the highest achievable nonlinearity for a quaternary function with two variables.

CHAPTER 3

Spectral Analysis of Rotation Symmetric Functions over Z_q and Extraction Of Rotation Symmetric Ternary Bent Functions With Reduced Computational Complexity

3.1 Introduction

In 1999, Pieprzyk and Qu [42] proposed a sub-class of Boolean functions named Rotation Symmetric Boolean Functions (RSBF). Cryptosystems avail the advantages of RSBF because of their desirable cryptographic properties. There are numerous applications for RSBF in cryptography. They are used in the design of fast hashing algorithms[42]. The extension of rotation symmetry to many-valued logic and the intriguing characteristics of these functions have received much discussion from authors in [49, 50]. The symmetricity of the rotation symmetric functions throughout its cyclic permutation allows representing these functions in their compact form so that the analysis becomes easy as it reduces the count of functions to

be examined. The compact structure of the rotation symmetric functions brings down the risk of spectral analysis and develops various methods to synthesise these functions [49]. An article [50] enunciates the bentness of the symmetric functions.

The number of q-functions in n variables becomes very large $(O(q^{q^n}))$ for higher values of n and q and hence the exhaustive search for finding the rotation symmetric function out of it is impractical. A formula for finding the count of RSBFs was given in[43]. The formula to find the count of rotation symmetric q-functions with the help of Burnside's lemma was provided in [45]. A mechanism for spectral analysis of rotation symmetric Boolean functions in its Walsh spectral domain [46] opens doors to an analogous method for functions of q-valued logic. This chapter is primarily concerned with presenting a mechanism for spectral analysis of rotation symmetric q-functions in its Vilenkin-Chrestenson transformants domain and, also to extract the complete class of rotation symmetric ternary bent functions in three variables by utilising this mechanism.

3.1.1 Rotation Symmetric Functions

Let us define a set of functions $\{\sigma_{n,i}, i = 1, 2...n\}$ on the set $\{x_1, x_2, ...x_n\}$ as follows,

$$\sigma_{n,i}\left(x_k\right) = x_{i+k(modn)} \tag{3.1.1}$$

Where modn is modulo n addition. Now, define a set of permutations on n symbols $\{x_1, x_2, ..., x_n\}$ by extending the function defined in 3.1.1,

$$\sigma_{n,i}\left(x_1, x_2, ..., x_n\right) = (\sigma_{n,i}(x_1), \sigma_{n,i}(x_2)..., \sigma_{n,i}(x_n)) \tag{3.1.2}$$

The permutation thus defined is a left circular shift of the vector $(x_1, x_2, ...x_n)$ through a position i.

Definition 3.1.1. *A function $f : V(n,q) \rightarrow Z_q$ is rotation symmetric if for any vector $(x_1, x_2, .., x_n) \in V(n,q)$ and $i = 1, 2..., n$, $f(\sigma_{n,i}(x_1, x_2, ..., x_n) = f(x_1, x_2, ..., x_n)$. In other words, if the function f is invariant for all cyclic permutations of its arguments, it is rotation symmetric.*

A rotation-symmetric ternary function (RSTF) in two variables has the following general structure:

x_1x_2	00	01	02	10	11	12	20	21	22
$f(x_1x_2)$	a	b	c	b	d	e	c	e	g

where $a, b, c, d, e, g \in Z_3$. Note that, $f(01) = f(10)$, $f(02) = f(20)$ and $f(12) = f(21)$, symmetric over all the cyclic rotations of (x_1, x_2).

Definition 3.1.2. *The orbit of an element $x \in V(n,q)$, denoted by O_x, is defined as the set of all cyclic rotations of x. That is, for $x \in V(n,q)$, the orbit is, $O_x = \{\sigma_{n,i}(x), i = 1, 2, ...n\}$.*

Remark 3.1.1. *The orbits form a partition of the set $V(n,q)$.*

For a two-variable ternary function, the orbits of its domain are, $O_{00} = \{00\}, O_{01} = \{01, 10\}, O_{02} = \{02, 20\}, O_{11} = \{11\}, O_{12} = \{12, 21\}$ and $O_{22} = \{22\}$.

The first element in orbit is known as the representative element when arranged lexicographically [46].

Now consider a ternary function in three variables whose truth table is, $f = [2\ 1\ 2\ 1\ 1\ 1\ 2\ 1\ 0\ 1\ 1\ 1\ 1\ 2\ 0\ 2\ 1\ 0\ 1\ 0\ 2\ 0\ 2\ 1]^T$.
The orbits of the domain, V(3,3) are, $O_{000} = \{000\}, O_{001} = \{001, 010, 100\}$, $O_{002} = \{002, 020, 200\}$, $O_{011} = \{011, 101, 110\}$, $O_{012} = \{012, 120, 201\}$, $O_{021} = \{021, 102, 210\}$, $O_{022} = \{022, 202, 220\}$, $O_{111} = \{111\}$, $O_{112} = \{112, 121, 211\}$, $O_{122} = \{122, 212, 221\}$ and, $O_{222} = \{222\}$.

We can see that the output of the function f at each element in an orbit has the same value and hence f is invariant under cyclic rotations. Therefore, f is rotation symmetric.

The count of the rotation symmetric functions is directly connected with the count of orbits of $V_(n,q)$. A formula to find the number of orbits of $V_(n,q)$ was given in [45];

$$o_n^q = \frac{1}{n}\sum_{d/n}\phi(d)q^{\frac{n}{d}} \tag{3.1.3}$$

Where $\phi(d)$ is the Euler-Totient function.

A rotation-symmetric function can be expressed in compressed form, which is made up of the function values of the representative elements. Since there are o_n^q representative elements, it is immediate that the count of n variable q- functions which are rotation symmetric is equal to $q^{o_n^q}$.

The table 3.1 compares the number of RSBFs and RSTF for various values of n.

Table 3.1: *Comparison of no.of RSBF and RSTF*

n	1	2	3	4	5	6	7
$No.of RSBFs$	2	2^3	2^4	2^6	2^8	2^{14}	2^{20}
$No.of RSTFs$	3	3^6	3^{11}	3^{24}	3^{51}	3^{130}	3^{315}

3.2 Spectral Analysis of Rotation Symmetric Functions over Z_q

The rotation symmetric functions are easier to analyse when they are represented in their compressed form. Let us designate R_i as the representative element of i^{th} orbit. Therefore, $R_1, R_2, ...R_{o_n^q}$ are the representative ele-

ments for a rotation symmetric function in n variables. It can be seen that the spectral coefficient of each element in an orbit is the same. A result concerning RSBF is provided in[46]. This result is modified [46] for Fourier transform of rotation symmetric functions over $GF(p)$ in [45]. Similar to [45] and also as the conclusion of the theorem 2 in[47], we have

Lemma 3.2.1. *Let $f : V(n,q) \rightarrow Z_q$ be a rotation symmetric q-function in n variables. Then every element in an orbit has the same Vilenkin-Chrestenson spectral coefficient. That is, $\Phi_f(u) = \Phi_f(v)$ if $u, v \in O_{R_i}$, $1 \leq i \leq o_n^q$.*

For example, consider a rotation symmetric ternary function in three variables whose truth table representation is $f = [10201020201010202120$ $2021212]^T$. Because of the spatial constraint, we present the vector of the absolute value of Vilenkin-Chrestenson spectral coefficients computed in accordance with expression in 2.2.13.

$\Phi_f = [3.00\ 3.00\ 5.19\ 3.00\ 5.19\ 3.00\ 5.19\ 3.00\ 3.00\ 3.00\ 5.19\ 3.00\ 5.19$ $15.87\ 7.90\ 5\ 7.90\ 0\ 5.19\ 3.00\ 3.00\ 3.00\ 7.90\ 0\ 3.00\ 0\ 3.00]$.

It is clear that,

$$
\begin{aligned}
|\phi_f(001)| = |\phi_f(010)| = |\phi_f(100)| &= 3.00 \\
|\phi_f(002)| = |\phi_f(020)| = |\phi_f(200)| &= 5.19 \\
|\phi_f(011)| = |\phi_f(101)| = |\phi_f(110)| &= 5.19 \\
|\phi_f(012)| = |\phi_f(120)| = |\phi_f(201)| &= 3.00 \\
|\phi_f(021)| = |\phi_f(102)| = |\phi_f(210)| &= 3.00 \\
|\phi_f(022)| = |\phi_f(202)| = |\phi_f(220)| &= 3.00 \\
|\phi_f(112)| = |\phi_f(121)| = |\phi_f(211)| &= 7.90 \\
|\phi_f(122)| = |\phi_f(212)| = |\phi_f(221)| &= 0.
\end{aligned}
$$

According to lemma 3.2.1 the spectral values of rotation symmetric functions are completely described by its representative elements.

Therefore, the Vilenkin-Chrestenson spectrum of such a function f is

$$\Phi_f = \{\Phi_f(R_1), \Phi_f(R_2), ..., \Phi_f(R_{O^n})\}.$$

Motivated from[46], let us construct a matrix $H = [h_{ij}]_{o_{nn}^q}^{q}$, where $h_{ij} = \sum_{x \in V_n} \omega^{\langle x, R_i \rangle}$.This matrix significantly reduces the amount of computation required for various analysis of rotation symmetric functions. The spectral values of a rotation symmetric q-function at each representative element can be determined using the following theorem.

Theorem 3.2.2. *Let $H = [h_{ij}]_{o_n^q \times o_n^q}$ with $h_{ij} = \sum_{x \in O_{R_i}} \omega^{\langle x, R_j \rangle}$.Then the spectral value of a representative element R_j is, $\Phi_f(R_j) = \sum_{i=1}^{o_n^q} F(R_i)\overline{h_{ij}}$. Where $\overline{h_{ij}}$ is the complex conjugate of h_{ij}.*

Proof.

$$\begin{aligned}
\Phi_f(R_j) &= \sum_{x \in V(n,q)} F(R_i)\overline{\omega^{\langle x, R_i \rangle}} \\
&= \sum_{i=1}^{o_n^q} \sum_{x \in O_{R_i}} F(R_i)\overline{\omega^{\langle x, R_i \rangle}} \\
&= \sum_{i=1}^{o_n^q} F(R_i) \sum_{x \in O_{R_i}} \overline{\omega^{\langle x, R_i \rangle}} \\
&= \sum_{i=1}^{o_n^q} F(R_i)\overline{h_{ij}}
\end{aligned}$$

□

By utilising the matrix H, one can determine the spectrum of f as follows.

$$\Phi_f = f^*.\overline{H}$$

Where f^* is the compressed form of the function f of length o_n^q that displays the function value at each of the representative elements.

It has been determined that for ternary functions with three variables, there are 11 orbits, 11 representative elements, and 11 components in the compressed form of an RSTF. In this instance, the matrix H is an 11×11. The computation of this matrix is performed using Python programming language. The complexity of the computation is reduced by approximately one-fifth when the matrix H is used to compute the spectrum of a ternary function in three variables.

$$
\mathrm{H} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 2+e^{\frac{2\pi}{3}i} & 2+e^{\frac{4\pi}{3}i} & 1+2e^{\frac{2\pi}{3}i} & 0 & 0 & 1+2e^{\frac{4\pi}{3}i} & 3e^{\frac{2\pi}{3}i} & 2e^{\frac{2\pi}{3}i}+e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i}+2e^{\frac{4\pi}{3}i} & 3e^{\frac{4\pi}{3}i} \\
3 & 2+e^{\frac{4\pi}{3}i} & 2+e^{\frac{2\pi}{3}i} & 1+e^{\frac{2\pi}{3}i} & 0 & 0 & 1+2e^{\frac{2\pi}{3}i} & 3e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i}+2e^{\frac{4\pi}{3}i} & 2e^{\frac{2\pi}{3}i}+e^{\frac{4\pi}{3}i} & 3e^{\frac{2\pi}{3}i} \\
3 & 1+2e^{\frac{2\pi}{3}i} & 1+2e^{\frac{4\pi}{3}i} & 2e^{\frac{2\pi}{3}i}+e^{\frac{4\pi}{3}i} & 0 & 0 & e^{\frac{2\pi}{3}i}+2e^{\frac{4\pi}{3}i} & 3e^{\frac{4\pi}{3}i} & 2+e^{\frac{4\pi}{3}i} & 2+e^{\frac{2\pi}{3}i} & 3e^{\frac{2\pi}{3}i} \\
3 & 0 & 0 & 0 & 3e^{\frac{4\pi}{3}i} & 3e^{\frac{2\pi}{3}i} & 0 & 3 & 0 & 0 & 3 \\
3 & 0 & 0 & 0 & 3e^{\frac{2\pi}{3}i} & 3e^{\frac{4\pi}{3}i} & 0 & 3 & 0 & 0 & 3 \\
3 & 1+2e^{\frac{4\pi}{3}i} & 1+2e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i}+2e^{\frac{4\pi}{3}i} & 0 & 0 & 2e^{\frac{2\pi}{3}i}+e^{\frac{4\pi}{3}i} & 3e^{\frac{2\pi}{3}i} & 2+e^{\frac{2\pi}{3}i} & 2e^{\frac{4\pi}{3}i} & 3e^{\frac{4\pi}{3}i} \\
1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 & 1 & e^{\frac{2\pi}{3}i} & 1 & e^{\frac{2\pi}{3}i} & e^{\frac{4\pi}{3}i} & 1 \\
3 & 2e^{\frac{2\pi}{3}i}+e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i}+2e^{\frac{4\pi}{3}i} & 2+e^{\frac{4\pi}{3}i} & 0 & 0 & 2+e^{\frac{2\pi}{3}i} & 3e^{\frac{2\pi}{3}i} & 1+2e^{\frac{2\pi}{3}i} & 1+2e^{\frac{2\pi}{3}i} & 3e^{\frac{4\pi}{3}i} \\
3 & e^{\frac{2\pi}{3}i}+2e^{\frac{4\pi}{3}i} & 2e^{\frac{2\pi}{3}i}+e^{\frac{4\pi}{3}i} & 2+e^{\frac{2\pi}{3}i} & 0 & 0 & 2+e^{\frac{4\pi}{3}i} & 3e^{\frac{4\pi}{3}i} & 1+2e^{\frac{2\pi}{3}i} & 1+2e^{\frac{4\pi}{3}i} & 3e^{\frac{2\pi}{3}i} \\
1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1 & 1 & e^{\frac{4\pi}{3}i} & 1 & e^{\frac{4\pi}{3}i} & e^{\frac{2\pi}{3}i} & 1
\end{bmatrix}
$$

3.3 Observations on Rotation Symmetric Ternary Functions in Two Variables

There are 729 rotation symmetric ternary functions in 2 variables. We converted these functions to their Reed-Muller form using the expressions in 2.2.10 and 2.2.11 to determine the algebraic degree and to analyse the nature of these functions. There are two homogeneous functions with algebraic degree 4, two with algebraic degree 3, eight with degree 2, and two with degree 1. Following [43], let us denote (n, m, k, l) be the functions with n variable, m-correlation immune, algebraic degree k and non-linearity l. The symbol " * " in a position denotes the property is not specified at the position. The computation enabled the collection of some cryptographically dominant functions which is depicted in table 3.2.

Table 3.2: *Cryptographic Characteristics of two Variable Rotation Symmetric Ternary Functions*

Nature of function	(n, m, k, l)	Number of Functions	Remark
Homogeneous	(2,*,2,6)	4	Bent functions
	(2,1,2,3.8)	2	
	(2,1,1,0)	2	Resilient functions
Non-Homogeneous	(2,*,2,6)	32	Bent functions
	(2,1,2,3.8)	6	
	(2,1,1,0)	6	Resilient functions

It is observed that each of the 1-resilient RSTF in two variables is of the degree 1 and it has the structure;

$$a_0 + a_1x_1 + a_2x_2$$

where $a_i \in \{0, 1, 2\}, i = 0, 1, 2$ with $a_1 = a_2 \neq 0$.

3.4 Method to Synthesise Rotation Symmetric Ternary Bent Functions

The extraction of rotation symmetric ternary bent functions (RSTBF) in three variables is the focus of this section. The challenge of handling a large number of functions for higher values of q is what drives the focus on the ternary function. Bent functions have special importance in cryptography. Studies on the symmetric Boolean bent function is carried out in[46] and that of ternary bent functions in two variables and partially on three variables have been conducted in[50]. A function is bent if each of its spectral values is a constant. Since a rotation symmetric function can be expressed by means of its representative elements, it is bent if the representative element in each orbit has uniform absolute spectral values. A ternary function in three variables is bent if the absolute values of the spectral coefficients satisfy, $|\Phi_f(R_i)| = 3^{\frac{n}{2}}$, $i = 1, 2..., o_n^3$ $\left(\sqrt{27}, if\ n = 3\right)$.

There are 486 bent functions when $n = 2$ [32] and the count is very large for ternary functions in three or more variables. In[33], a method to synthesise 3-variable ternary bent functions was discussed. An attempt was made to find the rotation symmetric bent functions in [50]. The authors found all 36 symmetric ternary bent functions in two variables and using the tensor sum method a lower bound for the rotation symmetric ternary bent functions in 3 variables was mentioned.

Bent q-functions possess some structural Characteristics[33]. This structural characteristic can be explained in terms of *structural weight*.

Definition 3.4.1. *Structural weight (sw) of a string with entries in Z_q is the n-tuple $[< n_0, n_1, ..., n_{q-1} >]$, where $n_i, i = 0, 1, ...q-1$ is the number of $i's$ in its truth table representation.*

The research in[33] tells us that the structural weights of ternary bent sequences when $n = 3$ are $[< 12, 9, 6 >], [< 9, 6, 12 >], [< 6, 12, 9 >]$, $[< 9, 12, 6 >], [< 12, 6, 9 >]$ and $[< 6, 9, 12 >]$. This structural characteristic is used to find the rotation-symmetric ternary bent functions in three variables. We made a computer search using Python programming language to extract the bent functions utilising the matrix H. A comprehensive search is performed for the rotation symmetric functions satisfying the sw criterion for bentness as given in [33], in its uncompressed form. First, we examined the functions with sw, $[< 12, 9, 6 >]$. The search carried over all functions with the given structure and calculated $\sum_{i=1}^{O_n^q} \omega^{f(R_i)\overline{h_{ij}}}, 0 \leq j \leq o_n^q$ and check the absolute value is $3^{n/2}$. If the value is not $3^{n/2}$ for some j, we discard the function and the search continues. It is found that the compressed form of the rotation symmetric ternary bent functions falls into two groups with sw $[< 6, 3, 2 >]$ and $[< 4, 5, 2 >]$. Further examination reveals that the other bent functions belong to the set of functions with sw, the permutations of these two structures, and can be generated by a set of

seed functions.

There are 36 seed functions with sw $[< 6, 3, 2 >]$.

$[00010110220]^T$ $[00011010220]^T$ $[[00020120110]^T$ $[00021020110]^T$
$[00100120120]^T$ $[00101020120]^T$ $[00120110200]^T$ $[00121010200]^T$
$[00200110210]^T$ $[00201010210]^T$ $[00210120100]^T$ $[00211020100]^T$
$[01010120020]^T$ $[01011020020]^T$ $[01020100210]^T$ $[01021000210]^T$
$[01100100220]^T$ $[01101000220]^T$ $01120120000]^T$ $[01121020000]^T$
$[01200120010]^T$ $[01201020010]^T$ $[01210100200]^T$ $[01211000200]^T$
$[02010100120]^T$ $[02011000120]^T$ $[02020110010]^T$ $[02021010010]^T$
$[02100110020]^T$ $[02101010020]^T$ $[02120100100]^T$ $[02121000100]^T$
$[02200100110]^T$ $[02201000110]^T$ $[02210110000]^T$ $[02211010000]^T$

The other seed functions belong to the set of functions with cardinality 18 and sw $[< 5, 2, 4 >]$, which are;

$[00012210220]^T$ $[00022220110]^T$ $[00102220120]^T$ $[00122210200]^T$
$[00202210210]^T$ $[00212220100]^T$ $[01012220020]^T$ $[01022200210]^T$
$[01102200220]^T$ $[01122220000]^T$ $[01202220010]^T$ $[01212200200]^T$
$[02012200120]^T$ $[02022210010]^T$ $[02102210020]^T$ $[02122200100]^T$
$[02202200110]^T$ $[02212210000]^T$//

Lemma 3.4.2. *[50] Let $f : V(n, q) \to Z_q$ be a rotation symmetric bent function, then $af + bI$, where $a < q$ and I is a row vector with all its component 1 and $b \in \{0, 1, ..., q-1\}$, are rotation symmetric bent functions. The addition is component-wise addition modulo q.*

Remark:- If q is a power of a prime number then Lemma3.4.2 holds for those values of a with $a < q$ such that $gcd(a, q) = 1$.

Using Lemma 3.4.2 we have $f, f+I, f+2I, 2f, 2f+I$ and $2f+2I$ are rotation symmetric ternary bent functions provided f is a rotation symmetric ternary bent functions and that each of the seed function generates 5 other rotation symmetric ternary bent functions. Thus we found 324 rotation symmetric ternary bent functions in 3 variables. The conversion to its original form can be obtained by reassigning the function value at each representative element to all elements in the respective orbits.

An algorithm for extracting the RSTBFs with the help of the matrix H is provided in Appendix F.

Rotation symmetric ternary bent functions follow 12 structural weight. Table3.3 gives the sw and the number of RSTBFs in a given structure.

Table 3.3: *Structural Characteristics of Rotation Symmetric Ternary Bent functions*

SW	No.of Functions	SW	No.of Functions
[<6, 3, 2 >]	36	[<5, 2, 4 >]	18
[<6, 2, 3 >]	36	[<5, 4, 2 >]	18
[<3, 6, 2 >]	36	[<4, 2, 5 >]	18
[<3, 2, 6 >]	36	[<4, 5, 2 >]	18
[<2, 3, 6 >]	36	[<2, 4, 5 >]	18
[<2, 6, 3 >]	36	[<2, 5, 4 >]	18

CHAPTER 4

Correlation Immunity of Quaternary Functions

4.1 Introduction

Boolean functions(BF) play a vital role in designing block ciphers and stream ciphers [64]. There are stream ciphers where the output from several linear feedback shift registers (LFSRs) is combined using a Boolean function to provide the keystream. The nonlinear BF for which, the inputs are predetermined stages of an LFSR generates the keystream as its output. Such stream ciphers are vulnerable to correlation attack[34]. Siegenthaler presented a correlation attack against stream ciphers in 1985[34]. Using the information of some keystream bits, the attacker tries to reconstruct the initial state of each individual LFSR independently. If the Boolean function is not properly chosen, it renders the cipher weak resistance to various attacks. There are many criteria to measure the quality of Boolean functions, among which nonlinearity and resistance against correlation attacks are predominant. The functions which have resistance against correlation attacks are known as correlation immune functions.

Various applications of the primitives based on many-valued logic confirm the need for further research in this direction. A great number of researches are devoted to developing cryptographic characteristics of non-binary functions and cryptographic constructions built upon them. The article [38] discusses the correlation immunity of ternary functions and the authors synthesised the complete class of correlation immune ternary functions in two variables. This chapter extends the correlation immunity to quaternary functions and presents a mathematical expression for the construction of quaternary resilient functions.

4.2 Correlation Immunity of Boolean Functions

Definition 4.2.1. [65] *Let f be a function in n independent and uniformly distributed random variables $x_1, x_2, \ldots .x_n$, the function $f\,(x_1, x_2, \ldots .x_n)$ is of m^{th} order, $(0 \leq m \leq n)$ correlation immunity if $f\,(x_1, x_2, \ldots .x_n)$ is independent of any set of m of its input variables, $x_1, x_2, \ldots, x_n$.*

In other words, f is of m^{th} order correlation immunity if its probability distribution is unaffected by any set of m input variables. That is, $Pr\,(f(x_1, x_2, ..., x_n)/x_{i_k} = s_k, 1 \leq k \leq m) = Pr\,(f(x_1, x_2, ..., x_n))$.

Definition 4.2.2. *A balanced correlation immune function is known as resilient.*

To define correlation immunity in terms of their sub-functions [65], let us recap the definition of the sub-function.

Definition 4.2.3. [65] *A sub-function of a Boolean function f is the function f^I derived from f by inserting constant values 0 or 1 for some of its variables.*

For an n variable function, if we substitute m of its variables by constants, the obtained sub-function is a function of $n - m$ variables.

Definition 4.2.4. [65] *A Boolean function on $V(n,q)$ is of m^{th} order, $(0 \leq m \leq n)$ correlation immunity if each of its sub-function of $n-m$ variable is of weight equal to $wt(f)/2^m$. That is, $wt(f^I) = wt(f)/2^m$.*

The correlation immunity of a Boolean function can also be characterised using Walsh transform as given by the following theorem.

Theorem 4.2.5. [66] *A Boolean function has correlation immunity of order m if and only if its Walsh transform is zero for all inputs with weight less than or equal to m. ie $W_f(u) = 0$, for $0 \leq wt(u) \leq m$.*

4.3 Correlation Immunity of Functions over Z_q

To define the correlation immunity of functions over Z_q, it is required to generalize the concept of sub-function. The sub-function of a q-function is defined as follows.

Definition 4.3.1. [38] *A sub-function of a q- function f is a function f^I derived from f by inserting constant values of the set $Z_q = \{0, 1, \ldots..q-1\}$ for some of its variables.*

The correlation immunity of the ternary function was defined in terms of the imbalance of the function[67]. The imbalance can be generalized for functions of q- valued logic as,

Definition 4.3.2. *The imbalance of the q- function is the absolute value of the first coefficient of the Vilenkin-Chrestenson transform. That is the absolute value of the sum of the component-wise product of the function by the sequence $[e^{i0}, e^{i0}, \ldots\ldots\ldots, e^{i0}]$.*

One can compute the imbalance of a q- function from its truth table representation.

let, n_k, k=0, 1, 2,...,$q-1$ are the number of ks in the q-function f . Then the imbalance of f is

$$\Delta_f = |\sum_{k=0}^{q-1} n_k e^{\frac{2\pi i}{q}k}| \tag{4.3.1}$$

Definition 4.3.3. *Let f be a q- function in n variables, f has correlation immunity of order m, $0 \leq m \leq n$ if the imbalance of any of its sub-functions f^I obtained by substituting m of its variable with constants from Z_q is, $\Delta_{f^I} = \Delta_f/q^m$.*

A q- function is m-resilient if it is correlation-immune and its truth table is uniformly distributed.

In line with the formal definition of the correlation immunity, the resiliency can be defined as follows,

Definition 4.3.4. *Let f be a q-function in n independent and uniformly distributed random variables $x_1, x_2, \ldots .x_n$, the function $f\,(x_1, x_2, \ldots .x_n)$ is m-resilient, $(0 \leq m \leq n)$ if $Pr\,(f(x_1, x_2, ..., x_n)/x_{i_k} = s_k, 1 \leq k \leq m) = Pr\,(f(x_1, x_2, ..., x_n)) = \frac{1}{q}$.*

4.4 Resilient Quaternary Functions

In order to analyse the resiliency, we make use imbalance of the functions. The imbalance of the quaternary function $f(x)$ can be computed using the following equation.

$$\Delta_f = |n_0.1 + n_1.i + n_2. - 1 + n_3. - i| \tag{4.4.1}$$

Where n_0, n_1, n_2 and n_3 are respectively the count of 0s, 1s, 2s and 3s in f when represented as a string of quaternary numbers.

Definition 4.4.1. *Let f is a 4- function in n variables, f is said to have m^{th} order correlation immunity, $0 \leq m \leq n$ if the imbalance of each of its sub-functions f^I of $n-m$ variables is, $\Delta_{f^I} = \Delta_f/4^m$.*

To illustrate the resiliency, let f be a quaternary function in two variables x_1 and x_2, whose truth table is given by

x_1x_2	00	01	02	03	10	11	12	13	20	21	22	23	30
$f(x_1,x_2)$	0	1	2	3	1	2	3	0	2	3	0	1	3

x_1x_2	31	32	33
$f(x_1,x_2)$	0	1	2

Simply, this function can be expressed as f =$[0123123023013012]^T$. Clearly, the imbalance of f is equal to zero. Now let us examine the nature of the sub-functions of f obtained by fixing one of its variables;

$$
\begin{aligned}
f(0,x_2) &= \{f(0,0)f(0,1)f(0,2)f(0,3)\} = \{0123\}\\
f(1,x_2) &= \{f(1,0)f(1,1)f(1,2)f(1,1)\} = \{1230\}\\
f(2,x_2) &= \{f(2,0)f(2,1)f(2,2)f(2,3)\} = \{2301\}\\
f(3,x_2) &= \{f(3,0)f(3,1)f(3,2)f(3,3)\} = \{3012\}\\
f(x_1,0) &= \{f(0,0)f(1,0)f(2,0)f(3,0)\} = \{0123\}\\
f(x_1,1) &= \{f(0,1)f(1,1)f(2,1)f(3,1)\} = \{1230\}\\
f(x_1,2) &= \{f(0,2)f(1,2)f(2,2)f(3,2)\} = \{2301\}\\
f(x_1,3) &= \{f(0,3)f(1,3)f(2,3)f(3,3)\} = \{3012\}
\end{aligned}
$$

Clearly, each of the sub-functions of one variable has zero imbalance and $\Delta_{f^I} = 0 = \frac{\Delta_f}{4^m}$. Thus, $f(x_1,x_2)$ is a 1-resilient function.

4.5 Method to Synthesise Resilient Quaternary Functions in Two Variables

The cardinality of the set of all quaternary functions in two variables is $4^{16} = 4294967296$. The extraction of resilient function out of this is infeasible from this large sample space by trial and error method. A subclass of quaternary resilient functions in two variables is synthesized in [68] and the extension to get the complete class of two variable quaternary 1-resilient functions is carried out by further research.

In this section, we present the formulation of a mathematical expression for the construction of two variable 1-resilient quaternary functions by employing permutation group action on a set consisting of four elements each of which is a string of length four whose arguments assume value from the set $Z_4 = \{0, 1, 2, 3\}$.

Definition 4.5.1. *Let G be a group and X be a set. Group action is a mapping from $G \times X$ to X that assigns each pair of elements $g \in X$ and $x \in X$ by a rule denoted by g.x satisfying the following axioms,*

1. *For every $g \in X$ and $x \in X, g.x \in X$*
2. *If $I \in G$ is the identity,then for every $x \in X, I.x$ =x*
3. *For every $g, h \in Gandx \in X, (g.h).x = g.(h.x)$*

Resilient functions are balanced, so we considered those functions and made a computer search on the reduced space for those that satisfy the condition for resiliency using Python programming language. This program could find all quaternary 1-resilient functions in two variables. The purpose of the research is to find a Mathematical expression for this class of functions. Detailed analysis of the output of the computer program

is performed and a unique mathematical expression for generating the quaternary resilient functions in two variables is obtained.

Consider the set $S = \{\mu_0, \mu_1, \mu_2, \mu_3\}$, where $\mu_0 = (0123), \mu_1 = \mu_0 + I$, $\mu_2 = \mu_0 + 2I, \mu_3 = \mu_0 + 3I$, where I is a string of length 4 with all its components equal to 1. The addition is performed component-wise modulo 4. Now, let us recall the symmetric group S_4 = $\{\gamma_i, i = 1, 2. . . . .4! : \gamma_i$, is a permutation on 4 symbols$\}$. Using the group action of the symmetric group S_4 on the set S and then by the concatenation of these group actions, one can generate a class of 144 1- resilient quaternary functions in two variables of length 16. It is now possible to generate the other 1-resilient functions using this class of 144 functions as the generating functions. The algebraic expression is as follows.

Consider the group action,

$$\psi : S_4 \times S \to S$$

$$\psi(\gamma_i, \mu_j) = \gamma_i(\mu_j), \quad i = 1, 2, ..., 24, \ , j = 0, 1, 2, 3. \tag{4.5.1}$$

Let $j, k, l, m \in \{0, 1, 2, 3\}$. Then for a fixed j such that $j \neq k \neq l \neq m$, the concatenation,

$$\gamma_i(\mu_j)||\gamma_i(\mu_k)\gamma_i(\mu_l)||\gamma_i(\mu_m), i = 1, 2, ..., 24, \tag{4.5.2}$$

produces the truth table of the 144 1- resilient quaternary functions in two variables. Each of the resilient functions produces another three 1-resilient quaternary functions of length 16, by applying a transformation g from the

set of quaternary sequences of length 16 to itself, defined as;

$$g(x_1x_2....x_{16}) = (y_1y_2....y_{16}), \quad where\ y_i = \begin{cases} k & if\ x_i = l \\ l & if\ x_i = k \\ x_i & otherwise \end{cases} \tag{4.5.3}$$

By replacing particular values for k and l we get the other quaternary resilient functions. The pair of values (k, l) that produce the required functions are (0,1),(0,3), and (2,3). We verified the resiliency of these functions using a Python programming language. Thus, from the method above we could form the complete class of 576 1- resilient quaternary functions of length 16 in two variables.

For example, consider the permutation $\gamma_i = (3241)$ and $\mu_j = (2301)$, $\mu_k = (1230), \mu_l = (0123)$ and $\mu_m = (3012)$. Then the proposed expression constructs the function $f = [1320021331022031]^T$. Clearly $\Delta_f = 0$. Now consider its sub-functions obtained after fixing one of its variables by the elements of the set $\{0123\}$, which are $\{1320\}, \{0213\}, \{3102\}, \{2031\}$, $\{1032\}, \{3210\}, \{2103\}$ and $\{0321\}$. It is to be noted that the imbalance of each of the sub-functions is equal to zero and hence the constructed function is 1-resilient.

The nonlinearity of the resilient functions obtained are computed using 2.2.15 and conversion of these functions to algebraic normal form could enable to find the algebraic degree with the help of Python programming language. Let k denote the algebraic degree and l denote nonlinearity. Then the table 4.1 depicts the number of functions with specific algebraic degree and nonlinearity. The resilient functions with $l = 9.67$ and $l = 8$ are of special importance and they can be used as primitives for various cryptographic applications to attain optimal results.

Table 4.1: *Algebraic Degree and Nonlinearity of the 1-resilient Functions in Two Variables*

No.of functions	$k=4$	$k=3$
$l=9.67$	112	16
$l=8.00$	184	120
$l=4.68$	80	48
$l=0.0$	8	8

Based on the result of the obtained two variable 1-resilient functions, a method for construction of quaternary 1-resilient functions in three variables was presented through a conjecture in[68]. Here, this conjecture is generalised to a theorem for m-resilient quaternary functions, a particular case of such construction in [37]

Theorem 4.5.2. *Let $f_0, f_1, f_2, f_3 : V(n,4) \to Z_4$ are m- resilient 4- functions in n variables, then for $x \in V(n,4)$, the $n+1$ variable 4-valued function $f : V(n+1,4) \to Z_4$ defined as*

$$f(x, x_{n+1}) = \begin{cases} f_0(x) & if \ x_{n+1} = 0 \\ f_1(x) & if \ x_{n+1} = 1 \\ f_2(x) & if \ x_{n+1} = 2 \\ f_3(x) & if \ x_{n+1} = 3 \end{cases}$$

is also an m-resilient quaternary function.

Proof. Let f_i, $i = 0, 1, 2, 3$ are m-resilient quaternary functions in n variables $x_1, x_2, ..., x_n$, where $x_i \in \{0, 1, 2, 3\}$. Let $x = (x_1, x_2, ..., x_n)$. Then we have for $k \in \{0, 1, 2, 3\}$, $1 \leq i_j \leq n$ and, $j = 1, 2, ..., m$

$$\begin{aligned} P\{f_i(x_1, x_2, ..., x_n) = k / (x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m)\} \\ = P\{f_i(x_1, x_2, ..., x_n) = k\} = \frac{1}{4}. \end{aligned} \tag{4.5.4}$$

Let x_{n+1} be as in the definition of f. Since f_i, $i = 0, 1, 2, 3$ are independent of the variable x_{n+1}, we have,

$$P\{f_i(x_1, x_2, ..., x_n) = k/\,(x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m, x_{n+1} = y_{n+1})\}$$
$$= P\{f_i\,(x_1, x_2, \ldots .x_n) = k\}/(x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m)\} \tag{4.5.5}$$

Also for $l \in \{0, 1, 2, 3\}$

$$P\{f_i\,(x_1, x_2, \ldots .x_n, x_{n+1}) = k\}/\,(x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m, x_{n+1} = l)\}$$
$$= P\{f_l\,(x_1, x_2, \ldots .x_n) = k/\,(x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m)\} = \frac{1}{4} \tag{4.5.6}$$

Now, 4.5.4 and 4.5.6 imply that,

$$P\{f\,(x_1, x_2, \ldots .x_n, x_{n+1}) = k\}/\,(x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m, x_{n+1} = y_{n+1})\}$$
$$= P\{f\,(x_1, x_2, \ldots .x_n, x_{n+1}) = k\} = \frac{1}{4} \tag{4.5.7}$$

Equation 4.5.7 is true for any of the m variables of $x_1, x_2, ...x_n, x_{n+1}$ and hence we have,

$$P\{f\,(x_1, x_2, \ldots .x_n, x_{n+1}) = k\}/\,(x_{i_1} = y_1, x_{i_2} = y_2, ..., x_{i_m} = y_m)\}$$
$$= P\{f\,(x_1, x_2, \ldots .x_n, x_{n+1}) = k\} = \frac{1}{4} \tag{4.5.8}$$

Thus f is m-resilient. □

For example, consider four 1-resilient quaternary functions in two variables x_1 and x_2 constructed by the method in section 4.5,
$f_0 = [0312213010233201]^T$, $f_1 = [1230310220130321]^T$
$f_2 = [1320021320313102]^T$, $f_3 = [2013132031020231]^T$

The truth table of the quaternary function f in 3-variables x_1, x_2 and x_3 as per the theorem 4.5.2 is
$f = [01123230132120032301112330120230122300012130331230302312$
$02031121]^T$
Now the sub-functions are;
$f(0, x_2, x_3) = \{0112323013212003\}$, $f(1, x_2, x_3) = \{2301112330120230\}$
$f(2, x_2, x_3) = \{1223000121303312\}$, $f(3, x_2, x_3) = \{3030231202031121\}$
$f(x_1, 0, x_3) = \{0112230112233030\}$, $f(x_1, 1, x_3) = \{3230112300012312\}$
$f(x_1, 2, x_3) = \{1321301221300203\}$, $f(x_1, 3, x_3) = \{2003023033121121\}$
$f(x_1, x_2, 0) = \{0312213010233201\}$, $f(x_1, x_2, 1) = \{1230310220130321\}$
$f(x_1, x_2, 2) = \{1320021320313102\}$, $f(x_1, x_2, 3) = \{2013132031020231\}$

Clearly f is balanced and hence $\Delta_f = 0$. Each of the sub-functions is also uniformly distributed and hence $\Delta_f^I = 0$. Therefore the function f satisfies the conditions for resiliency.

If f is a q-valued resilient function, it is balanced. Since a circular shift or scaling by an element co-prime to q does not affect the balancedness of the function f and its sub-functions.This fact substantiates the following proposition.

Proposition 4.5.3. *Let $f : V(n, q) \to Z_q$ be a resilient function. Then $\{af + bI, a < q, gcd(a, q) = 1, b \in Z_q$, where I is a string of length q^n with all its entries 1, are resilient functions.*

The operations specified in proposition 4.5.3 are spectral invariant [69].

4.6 Method for Analysis of Resilient Functions Based on Orthogonal Matrix

A method for analysing the correlation immunity of functions over Z_q on the basis of an orthogonal matrix is conducted in [37]. This section is devoted to presenting similar arguments for the analysis of the relation between orthogonal matrix and correlation immunity of q-functions in terms of sub-functions.

Definition 4.6.1. [37] *An $L \times n$ matrix over Z_q is called an (L, n, q, m) orthogonal matrix if for any fixed m columns, each row vector $y \in V(m, q)$ appears exactly L/q^m times in the matrix consists of these m columns.*

Let $f : V(n, q) \to Z_q$ be an n variable q-function. For each j, $0 \leq j \leq q-1$, consider the matrix B_j with rows, the members of the set defined by $W_j = \{x \in V(n, q)\} : f(x) = j\}$. The cardinality of this set is n_j, the number of j's in the truth table of $f(x)$. Then B_j is a matrix of order $n_j \times n$. We have, $f(x)$ is m-correlation immune if the number of j's, $0 \leq j \leq q-1$, in each of the sub-functions of $n - m$ variables is equal to $\frac{n_j}{q^m}$. Since in that case, the imbalance of the sub-function f^I is;

$$\Delta_{f^I} = \left| \sum_{k=0}^{q-1} \frac{n_k}{q^m} e^{\frac{2\pi i}{q} k} \right| = \frac{1}{q^m} \Delta_f$$

The sub-function is obtained as a result of fixing m input variables and B_j consists of n_j rows, it is true that if we fix m columns of B_j, each vector of $V(m, q)$ appears exactly $\frac{n_j}{q^m}$ times. Also, for a balanced function $n_j = q^{n-1}$. From these arguments, if an n-variable q-valued function has resiliency of order m then B_j is a (q^{n-1}, n, q, m) orthogonal matrix, $0 \leq j \leq q-1$, the conclusion as in[37]

Thus, for a resilient function, the matrices B_j,$0 \leq j \leq q-1$ contain an equal number of rows, and if we fix m columns, every vector of $V(m,q)$ appears q^{n-m-1} times.

To illustrate the above discussion of the relation between resiliency of a quaternary function with the help of orthogonal matrices, consider a 2-resilient quaternary function in three variables whose truth table with input values is

$x_1x_2x_3$	000	001	002	003	010	011	012	013	020	021	022
$f(x_1,x_2,x_3)$	0	1	2	3	1	2	3	0	2	3	0

$x_1x_2x_3$	023	030	031	032	033	100	101	102	103	110	111
$f(x_1,x_2,x_3)$	1	3	0	1	2	1	2	3	0	2	3

$x_1x_2x_3$	112	113	120	121	122	123	130	131	132	133	200
$f(x_1,x_2,x_3)$	0	1	3	0	1	2	0	1	2	3	2

$x_1x_2x_3$	201	202	203	210	211	212	213	220	221	222	223
$f(x_1,x_2,x_3)$	3	0	1	3	0	1	2	0	1	2	3

$x_1x_2x_3$	230	231	232	233	300	301	302	303	310	311	312
$f(x_1,x_2,x_3)$	1	2	3	0	3	0	1	2	0	1	2

$x_1x_2x_3$	313	320	321	322	323	330	331	332	333
$f(x_1,x_2,x_3)$	3	1	2	3	0	2	3	0	1

The matrices B_0, B_1, B_2 and B_3 as per the above discussion are,

$$B_0=\begin{bmatrix}0&0&0\\0&1&3\\0&2&2\\0&3&1\\1&0&3\\1&1&2\\1&2&1\\1&3&0\\2&0&2\\2&1&1\\2&2&0\\2&3&3\\3&0&1\\3&1&0\\3&2&3\\3&3&2\end{bmatrix}, B_1=\begin{bmatrix}0&0&1\\0&1&0\\0&2&3\\0&3&2\\1&0&0\\1&1&3\\1&2&2\\1&3&1\\2&0&3\\2&1&2\\2&2&1\\2&3&0\\3&0&2\\3&1&1\\3&2&0\\3&3&3\end{bmatrix}, B_2=\begin{bmatrix}0&0&2\\0&1&1\\0&2&0\\0&3&3\\1&0&1\\1&1&0\\1&2&3\\1&3&2\\2&0&0\\2&1&3\\2&2&2\\2&3&1\\3&0&3\\3&1&2\\3&2&1\\3&3&0\end{bmatrix}, B_3=\begin{bmatrix}0&0&3\\0&1&2\\0&2&1\\0&3&0\\1&0&2\\1&1&1\\1&2&0\\1&3&3\\2&0&1\\2&1&0\\2&2&3\\2&3&2\\3&0&0\\3&1&3\\3&2&2\\3&3&1\end{bmatrix}$$

From the matrices, it is clear that $n_0 = n_1 = n_2 = n_3 = 16$ and if we fix any two rows of each of the matrices, every vector $\alpha \in V(2,4)$ appears exactly $n_j/4^2$ $(= 4^{3-2-1})$ times, $0 \leq j \leq 3$. Thus, B_j, $j = 0, 1, 2, 3$ are orthogonal matrices.

CHAPTER 5

New Design of S-box based on Galois Field of Odd Characteristic and Analysis of its Cryptographic Properties

5.1 Introduction

The construction of conventional cryptographic schemes solely believes in the two basic properties: confusion and diffusion introduced by Shannon[3]. Confusion makes the relation between the input and output as complex as possible by hiding the algebraic structure in the design. The primary objective of confusion is to make it difficult to identify the key, even if one has a sufficient collection of plaintext-ciphertext pairs generated with the same key. Accordingly, each bit of the ciphertext should be based on the entire key and in multiple ways on a number of bits of the key; changing one bit of the key must result in a complete change to the ciphertext. In ciphers, it is accomplished by substitution operation, especially by the application of S-boxes. Diffusion spreads out the minor change in the input

all over the output. The unique feature of diffusion is the dissipation of repetition in the plaintext statistics in the ciphertext statistics. In diffusion, the output bits should depend on the input bits in a challenging way, so that if the plaintext is altered by only one bit, the ciphertext should totally change in an unpredictable or pseudorandom way. Diffusion is achieved by performing the permutation transformations.

S-boxes are the core of the symmetric block algorithms that carry out the substitution. They are the nonlinear components of block ciphers that employ substitution permutation networks (SPNs), such as AES and DES. Over many years, their study received a great deal of interest. Substitutions must be carefully performed, as evidenced by the evolution of various security threats. Numerous strategies are suggested to make the S-box stronger and more difficult to attack.

The emergence of cryptographic algorithms for data security has been facilitated by the development of the theory of many-valued logic [55]. The block symmetric algorithm on ternary logic designed by Artem Sokolov et al effectively performs substitution operations by means of S-boxes [55]. So in order to build algorithms based on many-valued logic, there is room for building cryptographically robust S-boxes.

The cryptographic primitive in our construction is based on ternary logic. It makes use of operations on the extension field $GF(3^n)$ of F_3. The Galois Field $GF(3^n)$ is constructed by choosing a primitive irreducible polynomial over F_3. The elements in $GF(3^n)$ can be expressed as a polynomial with coefficients 0,1, or 2 and also as a string of ternary numbers.

5.1.1 Cryptographic properties of S-box

An S-box is the only nonlinear component in most of the well-known block ciphers. Therefore, the primary purpose of an S-box in an encryption scheme is to improve the nonlinear characteristics. The Nonlinearity of

an S-box can be computed from that of its component functions with which it is constructed. The nonlinearity of component q- function can be computed from its Vilenkin-Chrestenson transform coefficient using the expression given in equation 2.2.15. The nonlinearity of the worst component q-function determines the nonlinearity of the S-box[22]. That is, The nonlinearity NL_S of the S-box is

$$NL_S = min\{NL_f, f \text{ is a component } q - \text{function}\} \tag{5.1.1}$$

In order to analyse the propagation characteristics of the S-box constructed on non-binary logic one may need an understanding of the derivative of the q-valued function.

Definition 5.1.1. *[17]The derivative $D_u(f(x))$ of a q-valued function $f(x)$ in the direction of a vector u is the q-valued function*

$$D_u(f(x)) = f(x \oplus_q u) - f(x)(modq) \tag{5.1.2}$$

Where $\oplus_q$ is the modulo q addition.

Definition 5.1.2. *[17] Let $f(x)$ be a q- function, $f(x)$ is said to satisfy the propagation criterion with respect to a vector u if the derivative in the direction of u is a balanced function. That is, if $n_0 = n_1 = ..., n_{q-1}$, where n_i is the number of i in its derivative. In other words, $Prob\{D_u(f(x)) = i\} = \frac{1}{q}$, for $i = 0, 1, 2, ..., q-1$.*

If the function $f(x)$ satisfies the propagation criterion with respect to all vectors u with $0 \leq wt(u) \leq m$, then it is said to have the propagation criterion of order m.

Definition 5.1.3. *[17] A function $f(x)$ of q-valued logic satisfies SAC if it satisfies propagation characteristics of order 1.*

In the design of the S-box, the input-output correlation minimization must receive the proper attention. The degree of statistical dependency of the S-box output on its input determines its input-output correlation. As the measure of the input-output correlation of the component q- functions, the input-output correlation coefficient ρ_{xy}, x, y=0, 1...$q-1$ and the correlation matrix P_{xy} consisting of the absolute values of the correlation coefficient was introduced in [41].

$$\rho_{xy} = \frac{\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2 - \left(\frac{\sum_{i=1}^{n} x_i}{n}\right)^2} \sqrt{\frac{1}{n}\sum_{i=1}^{n} {y_i}^2 - \left(\frac{\sum_{i=1}^{n} y_i}{n}\right)^2}} \quad (5.1.3)$$

5.2 Design principle of S-boxes

An S-box with a Galois field based design can be identified as a permutation of the elements in the Galois field that is used for its construction. Thus, one may view the mechanism to construct an S-box as finding a permutation that satisfies almost all characteristics that are expected for an ideal S-box. The design of the proposed S-box is adapted from AES S-box [2, 4] which follows Nyberg design [70] combined with an affine map of the form $S(x) = Ax^{-1} + B$, A is an invertible 8 × 8 matrix and B is 8 × 1 matrix over F_2. In our construction, we used the algebraic structure $S(x) = Ax^7 + B$, where A is an invertible matrix and B is a column matrix over F_3 with suitable order. The matrices are chosen in such a way that they minimise the computation. The shift matrix B assures there are no fixed points. The power mapping x^7 modulo the irreducible polynomial gives a bijective map over all finite field domains that we used for construction. We constructed four bijective S-boxes of sizes 9, 27, 81, and 243 and analysed their cryptographic characteristics such as the avalanche effect, nonlinearity, imbalance, and input-output correlation.

5.3 S-box of length 9

The constructed S-box of size 9 is a permutation of elements in the Galois field $GF(3^2)$. The field is constructed by choosing a primitive irreducible polynomial $p(x) = x^2 + x + 2$ over F_2 of order two. So the elements are polynomials of degree 1 with coefficients in F_3 which in turn can be described as a string of ternary numbers as $f_1 f_2$ consisting of the coefficients of these polynomials. In our construction, we used the non-degenerate matrix A and the shift matrix B as follows.

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \quad and \quad B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

The resulting S-box has two ternary components as given in the table 5.1.

Table 5.1: *S-box of length 9*

($x_1 x_0$)	00	01	02	10	11	12	20	21	22
S=$f_1(x_1, x_0) f_2(x_1, x_0)$	12	10	11	22	21	01	02	20	00

This S-box can also be depicted as a set of ternary strings of length two as, $S_9 = \{12\ 10\ 11\ 22\ 21\ 01\ 02\ 20\ 00\}$.

In order to analyse the avalanche characteristics of the proposed S-box, the derivatives of each of the component ternary functions in the direction of the vectors 01,10,20, and 02 of weight 1 are calculated and it is depicted in table 5.2

The table 5.3 is the measure of the probability of the number of 0's, 1's, and 2's in the derivatives of the component 3-functions. A ternary S-box of length 3^n satisfies the strict avalanche criterion if this probability value is equal to 0.33. In order to quantify the measure of SAC, the standard deviation of these probability values from the expected value was computed.

Table 5.2: *Derivatives of Component Functions*

D_{01}	f_1	0	0	0	0	1	2	2	1	0
	f_2	1	1	1	2	0	1	1	0	2
D_{10}	f	1	1	2	1	0	0	1	2	1
	f_2	0	1	0	0	2	2	0	0	1
D_{02}	f_1	0	0	0	1	0	2	0	1	2
	f_2	2	2	2	2	1	0	1	2	0
D_{20}	f_1	2	1	2	2	2	1	2	0	0
	f_2	0	0	2	0	2	0	0	1	1

Table 5.3: *Measures Propagation Characteristics of S-box 9*

u	01		10		02		20	
Components	f1	f2	f1	f2	f1	f2	f1	f2
Pr(0's)	0.56	0.22	0.22	0.56	0.56	0.22	0.22	0.56
Pr(1's)	0.22	0.56	0.56	0.22	0.22	0.22	0.22	0.22
Pr(2's)	0.22	0.22	0.22	0.22	0.22	0.56	0.56	0.22

The values deviate from the expected value with a standard deviation of 0.16.

The correlation of output to the input of the S-box must be the minimum for optimal results. As the measure of input-output correlation, the matrix of input-output correlation coefficients is calculated using the expression 5.1.1. The matrix for the S-box of size 9 is,

$$P_1 = \begin{bmatrix} 0.2 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

The nonlinearity (NL) of the constructed S-box is computed using the expression 2.2.15 and it found that the nonlinearity NL= 3.

5.4 S-box of length 27

The construction of the S-box of order 27 utilise the Galois field constructed using the primitive irreducible polynomial $p(x) = x^3 + 2x^2 + 1$ of order 3 over F_3. The elements in the field are polynomials of degree at most three.

consequently, the ternary representation has three components $f_1 f_2 f_3$. The decimal representation of these field elements contains numbers from 0 to 26. In this construction, we used the invertible matrix A and shift vector C as follows.

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix} and \, \mathrm{C} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

The constructed S-box in its ternary representation is depicted as the set S_{27}. One can perform substitution transformation by mapping the ternary blocks from 000 to 222 (0 to 26) respectively by the 27 ternary blocks in S_{27} in the given order.

$S_{27} = \{120 \;\; 210 \;\; 000 \;\; 020 \;\; 222 \;\; 212 \;\; 220 \;\; 001 \;\; 021 \;\; 221 \;\; 122 \;\; 201 \;\; 110 \;\; 011 \;\; 211 \;\; 111 \;\; 112 \;\; 200 \;\; 022 \;\; 012 \;\; 121 \;\; 102 \;\; 010 \;\; 101 \;\; 100 \;\; 002 \;\; 202\}$.

The propagation Characteristics (avalanche effect) of the component 3-functions of the S-box are computed using their derivatives at each vector of weight 1, the probability of 0's, 1's and 2's in its derivatives of each of the component functions is given in the table 5.4. These values come close to the expected value of 0.33 for an ideal S-box. The standard deviation of the values from the expected value is 0.09. The nonlinearity of the S-box is NL=18.031, higher than the NL=11.412, of the S-box constructed by Kim's Scheme[16]. The input-output correlation matrix is,

$$P_1 = \begin{bmatrix} 0.17 & 0 & 0.33 \\ 0 & 0.33 & 0 \\ 0.17 & 0.33 & 0.17 \end{bmatrix}$$

It is observed that there are three 0's and all other values are below 0.5 with a maximum of 0.33. Each component 3-function is balanced.

Table 5.4: *Measures of Avalanche Characteristics of S-box 27*

u	Components	0's	1's	2's
001	f_1	0.2222	0.3333	0.4444
	f_2	0.4444	0.2222	0.3333
	f_3	0.3333	0.4444	0.2222
010	f_1	0.4444	0.2222	0.3333
	f_2	0.3333	0.4444	0.2222
	f_3	0.3333	0.2222	0.4444
100	f_1	0.2222	0.3333	0.4444
	f_2	0.3333	0.2222	0.4444
	f_3	0.2222	0.3333	0.4444
002	f_1	0.2222	0.4444	0.3333
	f_2	0.4444	0.3333	0.2222
	f_3	0.3333	0.2222	0.4444
020	f_1	0.4444	0.3333	0.2222
	f_2	0.3333	0.2222	0.4444
	f_3	0.3333	0.4444	0.2222
200	f_1	0.2222	0.4444	0.3333
	f_2	0.3333	0.4444	0.2222
	f_3	0.2222	0.4444	0.3333

5.5 S-box of length 81

The construction of the S-box of order 81 utilises algebraic manipulations over the Galois field $GF(3^4)$ to produce a permutation of the elements in the field. The field $GF(3^4)$ is constructed as an extension field of F_3 by choosing an irreducible polynomial $p(x) = x^4 + x^3 + 2$ of degree 4 over F_3. So the elements are polynomials of degree at most 3. The 4×4 invertible matrix A and the shift matrix C used for the construction is

$$A = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 1 & 2 & 0 & 1 \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix}$$

Since the elements of $GF(3^4)$ are polynomials of degree at most three, the four coefficients of these polynomials derive the S-box, and hence each block in the S-box has four components. The S-box length 81 thus constructed is

$S_{81} = \{1021\ \ 1121\ \ 1200\ 2010\ \ 0202\ \ 2012\ 0002\ \ 0000\ \ 2110\ 0021$
$0200\ 1022\ \ 2212\ \ 0221\ 0001\ \ 1222\ \ 1012\ 1121\ \ 2021\ \ 1020\ 2112\ \ 1120$
$1221\ \ 1000\ \ 0100\ 2011\ \ 2121\ \ 2000\ 1111\ \ 0210\ \ 1011\ 0122\ \ 1112\ \ 0120$
$0212\ \ 2101\ \ 1110\ 1122\ \ 2002\ \ 2221\ 2202\ \ 1002\ \ 2222\ 1210\ \ 2211\ \ 0011$
$0220\ \ 2210\ \ 2201\ 2111\ \ 2020\ \ 1100\ 2022\ \ 1101\ \ 0012\ 2102\ \ 1201\ \ 2222$
$0211\ \ 2100\ \ 1001\ 2200\ \ 2220\ \ 2001\ 0102\ \ 2122\ \ 1212\ 1211\ \ 0020\ \ 0111$
$0022\ \ 0201\ \ 1202\ \ 0010\ \ 1220\ \ 2120\ \ 0101\ \ 1102\ \ 0121\ \ 1010\ \ 0110\}$

As the measure of the avalanche effect, the probability of 0's, 1's, and 2' of each of the four components is computed in the table 5.5. The values get sharpened to the expected value of 0. 33. The values deviate from the expected value with a standard deviation of 0.0663. The input-output correlation matrix P_3, of the S-box gives a positive sign as the maximum value of the correlation coefficient is 0.2. The nonlinearity of the S- box is $NL = 54.012$, much bigger than the $NL = 34.235$ of the s box of size 81 in [16]. Each of the components of the S-box is balanced also.

$$P_3 = \begin{bmatrix} 0.1 & 0.2 & 0.1 & 0 \\ 0 & 0.1 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

Table 5.5: *Avalanche Characteristics of S-box 81*

u	Components	Prob(0's)	Prob(1's)	Prob(2's)
0001	f_1	0.407407	0.296296	0.296296
	f_2	0.296296	0.407407	0.296296
	f_3	0.234568	0.45679	0.308642
	f_4	0.259259	0.308642	0.432099
0100	f_1	0.234568	0.419753	0.345679
	f_2	0.407407	0.296296	0.296296
	f_3	0.271605	0.382716	0.345679
	f_4	0.395062	0.320988	0.283951
0010	f_1	0.209877	0.358025	0.432099
	f_2	0.259259	0.37037	0.37037
	f_3	0.222222	0.444444	0.333333
	f_4	0.234568	0.45679	0.308642
1000	f_1	0.296296	0.407407	0.296296
	f_2	0.555556	0.222222	0.222222
	f_3	0.209877	0.308642	0.481481
	f	0.345679	0.382716	0.271605
0002	f_1	0.407407	0.296296	0.296296
	f_2	0.296296	0.296296	0.407407
	f_3	0.234568	0.308642	0.45679
	f_4	0.246914	0.432099	0.320988
0020	f_1	0.209877	0.407407	0.382716
	f_2	0.259259	0.358025	0.382716
	f_3	0.234568	0.320988	0.444444
	f_4	0.234568	0.308642	0.45679
0200	f_1	0.234568	0.345679	0.419753
	f_2	0.407407	0.296296	0.296296
	f_3	0.271605	0.37037	0.358025
	f_4	0.395062	0.283951	0.320988
2000	f_1	0.296296	0.296296	0.407407
	f_2	0.555556	0.222222	0.222222
	f_3	0.209877	0.469136	0.320988
	f_4	0.345679	0.283951	0.37037

5.6 S-box of length 243

S-box of length 243 utilises Galois field $GF(3^5)$ for its construction. The application of the transformation $S(x)$ to the elements of the field $GF(3^5)$ produces a permutation of the members of the same field. The irreducible polynomial $p(x) = x^5 + x^3 + 2x^2 + 1$ over F_3 is used for field formation. Thus, the members of the field $GF(3^5)$ are polynomials of degree at most 4. These polynomials when represented as a string of ternary numbers have five components. Consequently, each block of the S-box also has five components. The non-degenerate matrix and the shift matrix used for construction is

$$A = \begin{bmatrix} 1 & 0 & 2 & 0 & 1 \\ 0 & 1 & 0 & 2 & 1 \\ 1 & 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 1 & 0 \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

The ternary representation of the S-box is depicted as the set S_{243}.

Cryptographic characteristics of the S-box are measured and it is found that the avalanche effect of the components is very close to that expected for an ideal S-box. The deviation of the estimated values from the expected value was reduced to 0.034. The input-output correlation measure of the components shows that the input-output correlation is reduced compared to other S-boxes. The values are very close to zero, so the output of the S-box is almost uncorrelated with the input. The nonlinearity of the S-box is computed according to formulae 2.2.15 and 5.1.1 and it is found that the nonlinearity of the S-box of size 243 is NL= 204.0. The components of the S-box of length 243 are balanced.

S_{243} = {10201 11102 02000 12002 20020 20010 11100 20122
00112 01000 10100 01011 12201 01111 21101 02012 12210 22220
12102 22121 10002 21120 11212 11222 11201 22001 22021 21021
20220 21110 22011 01210 11120 10102 10200 10011 20110 10112
01201 01222 02202 11001 21000 02202 12222 00020 21100 22111
00202 11010 21021 01020 00010 20202 02120 22022 00212 00000
00121 10202 01121 12012 02222 20112 01021 02002 22112 00200
22112 00200 00122 20200 00022 22201 10020 02102 11210 21200
22210 12101 22200 22000 22100 20120 21111 01100 11011 10220
01012 01022 12021 11012 20001 21010 00210 11110 02210 12200
10222 22211 00110 10121 10101 21202 10111 02112 11020 00011
20102 12001 21102 02100 21122 22012 21221 11211 02101 02001
10010 00002 12220 10012 22110 21112 20222 12010 12211 22020
00220 02011 02220 02222 10021 01221 01220 00100 01112 21002
00001 21001 20211 00111 02010 11121 22102 12110 10110 21202
11220 11200 02122 20000 20201 20221 12221 11021 10211 02212
20012 01200 01110 12121 01102 00012 01002 10212 02110 22120
02021 02121 22002 11221 10011 20022 21020 20121 12112 10001
00021 02200 11111 00101 12120 21222 10210 12112 22122 12022
00222 20002 12100 12020 20021 12011 01122 20101 00221 22101
00211 12111 21211 12202 21011 02022 10221 00120 21220 11000
10022 11022 01010 00201 00102 02201 22202 12212 10000 12000
11101 02211 01001 02221 11002 01120 22010 01221 20212 11112
20111 22212 22221 21121 21210 21212 01101 20100 10122 02020
11122 20210 01212 21022 10120}

Table 5.6: *Measure of Propagation Characteristics of S-box 243*

u	Components	Pr(0's)	Pr(1's)	Pr(2's)
00001	f1	0.292181	0.353909	0.353909
	f2	0.37037	0.259259	0.37037
	f3	0.37037	0.296296	0.333333
	f4	0.333333	0.296296	0.37037
	f5	0.37037	0.333333	0.296296
00010	f1	0.358025	0.296296	0.345679
	f2	0.296296	0.296296	0.407407
	f3	0.296296	0.407407	0.296296
	f4	0.259259	0.37037	0.37037
	f5	0.37037	0.333333	0.296296
00100	f1	0.304527	0.341564	0.353909
	f2	0.37037	0.259259	0.37037
	f3	0.296296	0.333333	0.37037
	f4	0.259259	0.37037	0.37037
	f5	0.296296	0.37037	0.333333
01000	f1	0.337449	0.349794	0.312757
	f2	0.329218	0.378601	0.292181
	f3	0.337449	0.27572	0.386831
	f4	0.382716	0.349794	0.26749
	f5	0.308642	0.366255	0.325103
10000	f1	0.341564	0.378601	0.279835
	f2	0.222222	0.444444	0.333333
	f3	0.37037	0.333333	0.296296
	f4	0.259259	0.37037	0.37037
	f5	0.296296	0.296296	0.407407
00002	f1	0.292181	0.353909	0.353909
	f2	0.37037	0.37037	0.259259
	f3	0.37037	0.333333	0.296296
	f4	0.333333	0.37037	0.296296
	f5	0.37037	0.296296	0.333333
00020	f1	0.345679	0.345679	0.308642
	f2	0.304527	0.395062	0.300412
	f3	0.283951	0.308642	0.407407
	f4	0.26749	0.374486	0.358025
	f5	0.36214	0.308642	0.329218

u	Components	Pr(0's)	Pr(1's)	Pr(2's)
00200	f1	0.329218	0.353909	0.316872
	f2	0.345679	0.378601	0.27572
	f3	0.279835	0.382716	0.337449
	f4	0.251029	0.366255	0.382716
	f5	0.300412	0.304527	0.395062
02000	f1	0.353909	0.341564	0.304527
	f2	0.37037	0.296296	0.333333
	f3	0.296296	0.37037	0.333333
	f4	0.37037	0.296296	0.333333
	f5	0.296296	0.37037	0.333333
20000	f1	0.341564	0.279835	0.378601
	f2	0.222222	0.333333	0.444444
	f3	0.37037	0.296296	0.333333
	f4	0.259259	0.37037	0.37037
	f5	0.296296	0.407407	0.296296

The nonlinearity values and avalanche characteristics of the S-boxes are depicted in the graphs 5.1 and 5.2 respectively. The graph 5.1 clearly shows that as the size of the S-box increases, the nonlinear properties of the created S-boxes increase drastically. The graph 5.2 demonstrates that as the size of the S-box increases, the tendency for the S-boxes to satisfy the propagation criterion increases.

The table 5.7 depicts the cryptographic characteristics of the constructed S-boxes.

Designing an S-box that satisfies all the cryptographic characteristics at the high end is impractical. Here we concentrated more on the nonlinearity. The Contstucted S-boxes show a high level of nonlinearity so that these S-boxes may be useful for various cryptographic construction on ternary logic.

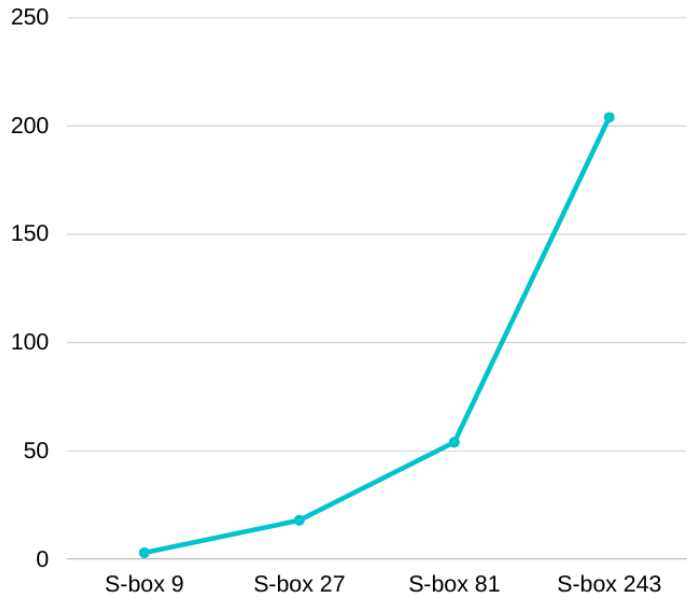

Figure 5.1: *Nonlinearity behavior of the proposed S-boxes*

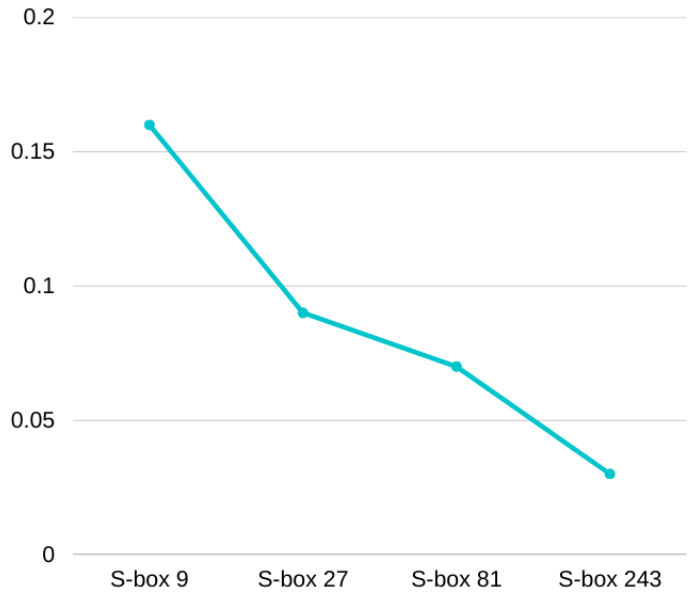

Figure 5.2: *Propagation Characteristics of the proposed S-boxes*

Table 5.7: *Comparison of Cryptographic Characteristics of Proposed S-boxes*

Size of S-box		9		27		81		243
		Proposed	In[16]	Proposed	In [16]	Proposed	In [16]	
Bijective		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Fixed points		No	Yes	No	Yes	No	Yes	No
Imbalance		0	0	0	0	0	0	0
Avalanche effect (Deviation from expected value)		0.16	0	0.09	0	0.07	0	0.03
Nonlinearity Distance (NL)		3	3	18.031	11.412	54.012	34.235	204
Input-output correlation	Min	0.2	0	0	0	0	0	0
	Max	0.7	0.5	0.33	0.5	0.2	0.5	0.2

CHAPTER 6

Conclusion

Cryptography has various applications. The construction, synthesis, and analysis of cryptographic primitives is an unending area of research. The need for secure cryptographic constructions is increasing with the development of technology. In this thesis, we focused on cryptographic primitives, functions over Z_q emphasising both ternary (3-functions) and quaternary (4-functions). The cryptographic properties, correlation immunity, resiliency, and Vilenkin-Chrestenson spectra of the functions of ternary and quaternary logic are extensively studied. A noval method for spectral analysis of Rotation symmetric q-functions is discussed and that could help to synthesise the rotation symmetric ternary bent functions in three variables with reduced computational complexity. The complexity could be reduced by one-fifth for the computation of the spectrum of rotation symmetric ternary functions in three variables. An indigenous method for synthesizing the resilient functions on quaternary logic is presented. The mathematical expression for the same is discussed. The relation between resiliency of

q-functions and orthogonal matrix in line with sub-functions is established . With the help of the Galois field of characteristic three, substitution boxes are constructed and analysed its cryptographic properties. These S-boxes possess a high level of nonlinearity and hence these S-boxes can be used in cryptographic algorithms based on ternary logic. The Python programming language is used for computation throughout the thesis.

CHAPTER 7

Recommendations and Future Directions

1. The computational efficiency is a constraint to move forward to the analysis of q-functions for the higher value of q and with more number of variables. So there is a scope to find theoretical methods to study the cryptographic properties of those functions.

2. Most of the images are represented in three components. The proposed S-boxes also contain three arguments 0, 1, and 2. So there is a scope to develop an iterative algorithm for image encryption utilising the constructed S-boxes.

3. Effective coding is the method for the transformation of binary code to some other domain in such a way that it would provide us with no unused combinations. Such a coding method could be constructed with the help of an arithmetic encoding algorithm. Such codes accelerate the applications of primitives on non-binary logic. So one can try to find codes transforming binary to q-valued logic, which can be utilized to develop primitives based on many-valued logic.

4. Most of the modern cryptographic algorithms can be represented in terms of 4-functions and 16-functions. So theoretical methods for cryptanalysis of those algorithms can be developed using many-valued logic.

Appendices

Appendix A: Algorithm for Converting Truth Table to ANF and Computing the Algebraic Degree(n=2)

```
import numpy as np
import math
# create the matrix of degree we want
def create_matrix(R3):
  n = 2
  for i in range(1,n):
    array_n=np.full((3**(i), 3**(i)), 3)
    array_0=np.full((3**(i), 3**(i)), 0)
row1 = np.hstack((R3, array_0, array_0))
row2 = np.hstack((array_0, np.mod(2*R3,
                   array_n np.mod(R3,array_n)))
row3 = np.hstack((np.mod(2*R3,array_n),
                  np.mod(2*R3,array_n),
                  np.mod(2*R3,array_n)))
    R3 = np.vstack((row1, row2, row3))
  return R3
def f_mul(f):
  R3=np.array([[1, 0, 0],
               [0, 2, 1],
               [2, 2, 2]])
  v=create_matrix(R3)
  if len(f)!=len(v):
    print('wrong dimensions for
```

```
                        matrix or f')
    return
  script = '¹²³'
  pol = ['xx', 'xx¹', 'xx²', 'x¹x', 'x¹x¹', 'x¹x²',
                          'x²x', 'x²x¹', 'x²x²']
  res = v@f
  res = np.array([i%3 for i in res])
  pol_final = ''
  for i in range(len(res)):
    pol_final+=(str(res[i])+pol[i])+' + '
  print('resultant matrix f*R9')
  print(res)
  print('polinomal: ')
  print(pol_final[:-2])
  return res
def result(v):
  n = int(math.sqrt(len(v)))
  v = np.reshape(v,(n,n))
  degree = 0
d={str(i):0 for i in range(len(v)+1,-1,-1)}
  for i in range(len(v)):
    for j in range(len(v)):
      if v[i][j]!=0:
        d[str(i+j)]=1
  for i in d:
    if d[i]!=0:
      degree = int(i)
      break
  if degree == 1 and d['0']==1:
    degree='affine'
  print('degree',degree)
 # input ternary function of legth 9.
f=[ ]
# multipy f with R9
v=f_mul(f)
#get degree from the resultant matrix
v=result(v)
```

Appendix B: Algorithm for Converting Truth Table to ANF and Computing the Algebraic Degree(n=3)

```
# general functions
import numpy as np
import math
# create the matrix of degree we want
def create_matrix(R3):
  n = 3
  for i in range(1,n):
    array_n=np.full((3**(i), 3**(i)), 3)
    array_0=np.full((3**(i), 3**(i)), 0)
    row1 = np.hstack((R3, array_0, array_0))
    row2 = np.hstack((array_0, np.mod(2*R3,array_n),
    np.mod(R3,array_n)))
    row3 = np.hstack((np.mod(2*R3,array_n),
    np.mod(2*R3,array_n), np.mod(2*R3,array_n)))
    R3 = np.vstack((row1, row2, row3))
  #print(len(R3))
  return R3
def f_mul(f):
  # initial matrix
  R3=np.array([[1, 0, 0],
               [0, 2, 1],
               [2, 2, 2]])
  v=create_matrix(R3)
  if len(f)!=len(v):
    print('wrong dimensions for matrix or f')
    return
  script = '¹²³'
  pol = ['xxx', 'xxx¹', 'xxx²', 'xx¹x',
  'xx¹x¹', 'xx¹x²', 'xx²x', 'xx²x¹', 'xx²x²',
  'x¹xx', 'x¹xx¹', 'x¹xx²', 'x¹x¹x', 'x¹x¹x¹',
  'x¹x¹x²', 'x¹x²x', 'x¹x²x¹', 'x¹x²x²', 'x²xx',
  'x²xx¹', 'x²xx²', 'x²x¹x', 'x²x¹x¹', 'x²x¹x²',
  'x²x²x', 'x²x²x¹', 'x²x²x²']
  res = v@f
  res = np.array([i%3 for i in res])
  pol_final = ''
  for i in range(len(res)):
    pol_final+=(str(res[i])+pol[i])+' + '
  print('resultant matrix f*R27')
  print(res)
```

```
  print('polinomal: ')
  print(pol_final[:-2])
  return res
def result(v):
  v = np.reshape(v,(3,3,3))
  degree = 0
  d={str(i):0 for i in range(6,-1,-1)}
  for i in range(len(v)):
    for j in range(len(v)):
      for k in range(len(v)):
        if v[i][j][k]!=0:
          d[str(i+j+k)]=1
  for i in d:
    if d[i]!=0:
      degree = int(i)
      break
  if degree == 1 and d['0']==1:
    degree='affine'
  print('degree',degree)
#Input ternary function of length 27
f=[ ]
# multipy f with R27
v=f_mul(f)
#get degree from the resultant matrix
v=result(v)
```

Appendix C: Algorithm for Converting Quaternary Functions to ANF and Computing Algebraic Degree (n=2)

```
import numpy as np
import math
# initial array
v4=np.array([[1, 0, 0, 0],
             [0, 1, 3, 2],
             [0, 1, 2, 3],
             [1, 1, 1, 1]])
# create the matrix of degree we want
def create_matrix(v4):
  n = 2
  mult_dict={(0,0):0,(0,1):0,(0,2):0,
    (0,3):0,(1,1):1,(1,2):2,(1,3):3,(2,2):3,(2,3):1,(3,3):2}
  v4_3=v4.copy()
  v4_2=v4.copy()
  for i in range(len(v4)):
    for j in range(len(v4)):
      v4_3[i][j]=mult_dict[(v4[i][j],3)]
  for i in range(len(v4)):
    for j in range(len(v4)):
      try:
        v4_2[i][j]=mult_dict[(v4[i][j],2)]
      except:
        v4_2[i][j]=mult_dict[2,(v4[i][j])]
  for i in range(1,n):
    array_n=np.full((4**(i), 4**(i)), 4)
    array_0=np.full((4**(i), 4**(i)), 0)
    row1 = np.hstack((v4, array_0, array_0, array_0))
    row2 = np.hstack((array_0, v4, v4_3, v4_2))
    row3 = np.hstack((array_0, v4, v4_2, v4_3))
    row4 = np.hstack((v4, v4, v4, v4))
    v4 = np.vstack((row1, row2, row3, row4))
  return v4
def f_mul(f):
  v4=np.array([[1, 0, 0, 0],
             [0, 1, 3, 2],
             [0, 1, 2, 3],
             [1, 1, 1, 1]])
  v=create_matrix(v4)
  if len(f)!=len(v):
```

```
    print('wrong dimensions for matrix or f')
    return
  #script = '¹²³'
  #pol = ['xx', 'xx¹', 'xx²', 'x¹x', 'x¹x¹', 'x¹x²',
                        'x²x', 'x²x¹', 'x²x²']
  res = v@f
  res = np.array([i%4 for i in res])
  #pol_final = ''
  #for i in range(len(res)):
  #  pol_final+=(str(res[i])+pol[i])+' + '
  # print('resultant matrix f*v9')
  # print(res)
 # print('polinomal: ')
 # print(pol_final[:-2])
  return res
def result(f):
  v=f_mul(f)
  n = int(math.sqrt(len(v)))
  v = np.reshape(v,(n,n))
  # print(v)
  degree = 0
  d={str(i):0 for i in range(6,-1,-1)}
  for i in range(len(v)):
    for j in range(len(v)):
      if v[i][j]!=0:
        d[str(i+j)]=1
  for i in d:
    if d[i]!=0:
      degree = int(i)
      break
  if degree == 1 and d['0']==1:
    degree='affine'
  return degree
```

Appendix D: Algorithm for Computation of Nonlinearity of Functions of Ternary Logic

```
 # general functions
import numpy as np
import math
# initial array
v3=np.array([[0, 0, 0],
             [0, 1, 2],
             [0, 2, 1]])
# enter the power of n we want to multiply with,
n=k
# create the matrix of degree we want
def create_matrix(n,v3):
  n = int(math.log(n,3))
  array_n=np.full((3**(n-1), 3**(n-1)), 3)
  for i in range(1,n):
    array_n=np.full((3**(i), 3**(i)), 3)
    row1 = np.hstack((v3, v3, v3))
    row2 = np.hstack((v3, np.mod(v3+1,array_n),
                                 np.mod(v3+2,array_n)))
    row3 = np.hstack((v3, np.mod(v3+2,array_n),
                                 np.mod(v3+1,array_n)))
    v3 = np.vstack((row1, row2, row3))
  return v3
# to transform matrix v to v_dash
def transform_matrix(v):
  n=len(v)
  for i in range(n):
    for j in range(n):
      if v[i][j]==1:
        v[i][j]=2
      elif v[i][j]==2:
        v[i][j]=1
  return v
import math
r3 = math.sqrt(3)
# to transform matrix v_dash to complex form
def transform_matrix_2(v):
  v_=v.copy()
  v_=v.astype(complex)
  n=len(v)
  for i in range(n):
```

```
    for j in range(n):
      if v_[i][j]==0:
        v_[i][j]=1
      elif v_[i][j]==1:
        v_[i][j]=(-1+r3*1j)/2
      elif v_[i][j]==2:
        v_[i][j]=(-1-r3*1j)/2
  return v_
  # f input
f=[]
# length of should be 3^n
def transform_f(f):
  for i in range(len(f)):
    if f[i]==0:
      f[i]=1
    elif f[i]==1:
      f[i]=(-1+r3*1j)/2
    elif f[i]==2:
      f[i]=(-1-r3*1j)/2
  print('this is f transformed: ')
  print(f)
  return f
def find_nf(f,v):
  if len(f)!=len(v):
    print('wrong dimension for matrix v and array f')
    return
  ln=len(f)
  res=f@v
  print(f'this is product of v{ln} and f')
  print([format(i, '.3f') for i in res])
  res=[abs(j) for j in res]
  print('this is modulus of f*v')
  print(res)
  res=max(res)
  nf=ln - res
  return nf
```

Appendix E: Algorithm for Computation of Nonlinearity of Functions of Quaternary Logic

```
import numpy as np
import math
# initial array
# enter the power of n
n=k
# create the matrix of degree we want
def create_matrix(n,v4):
  n = int(math.log(n,4))
  array_n=np.full((4**(n-1), 4**(n-1)), 4)
  for i in range(1,n):
    array_n=np.full((4**(i), 4**(i)), 4)
    row1 = np.hstack((v4, v4, v4, v4))
    row2 = np.hstack((v4, np.mod(v4+1,array_n),
                  np.mod(v4+2,array_n), np.mod(v4+3,array_n)))
    row3 = np.hstack((v4, np.mod(v4+2,array_n), v4,
                                          np.mod(v4+2,array_n)))
    row4 = np.hstack((v4, np.mod(v4+3,array_n),
                np.mod(v4+2,array_n), np.mod(v4+1,array_n)))
    v4 = np.vstack((row1, row2, row3, row4))
  return v4
# to transform matrix v to v_dash
def transform_matrix(v):
  n=len(v)
  for i in range(n):
    for j in range(n):
      if v[i][j]==1:
        v[i][j]=3
      elif v[i][j]==3:
        v[i][j]=1
  return v
import math
r3 = math.sqrt(3)
# don't need this here
# to transform matrix v_dash to complex form
def transform_matrix_2(v):
  v_=v.copy()
  v_=v.astype(complex)
  n=len(v)
    for j in range(n):
      if v_[i][j]==0:
```

```
        v_[i][j]=1
      elif v_[i][j]==1:
        v_[i][j]=1j
      elif v_[i][j]==2:
        v_[i][j]=-1
      elif v_[i][j]==3:
        v_[i][j]=-1j
  return v_
# F input
f=[]
# length of should be 3^n
def transform_f(f):
  for i in range(len(f)):
    if f[i]==0:
      f[i]=1
    elif f[i]==1:
      f[i]=1j
    elif f[i]==2:
      f[i]=-1
    elif f[i]==3:
      f[i]=-1j
  print('this is f transformed: ')
  print(f)
  return f
def find_nf(f,v):
  if len(f)!=len(v):
    print('wrong dimention for matrix v and array f')
    return
  ln=len(f)
  res=f@v
  print(f'this is product of f and v{ln}')
  print([format(i, '.3f') for i in res])
  res=[abs(j) for j in res]
  print('this is modulus of f*v')
  print(res)
  res=max(res)
  nf=ln - res
  return nf
```

```
def matrix_manipulation(f):
  n=16
  v4=np.array([[0, 0, 0, 0],
               [0, 1, 2, 3],
               [0, 2, 0, 2],
               [0, 3, 2, 1]])
  vn=create_matrix_2(n,v4)
  vn=transform_matrix(vn)
  vn=transform_matrix_2(vn)
  f=transform_f(f)
  nf = find_nf(f,vn)
  return round(nf,4)
f=[0, 1, 2, 2, 1, 0, 3, 1, 2, 3, 0, 1, 0, 2, 1, 0]
print('nf: ', matrix_manipulation(f))
```

Appendix F: Algorithm for Extraction of Rotation Symmetric Ternary Bent Functions

```
from iter tools import product
import numpy as np
# Check that the reduced array is equivalent to array containing
# 12 zero, 9 one, 6 two
def check_array_11(a):
  c={0:0,1:0,2:0}
  for i in a[:-3]:
    c[i]+=3
  for i in a[-3:]:
    c[i]+=1
  if c[0]!=12 or c[1]!=9 or c[2]!=6:
    return False
  return True
# Define the elements whose combination needed
elements = [0, 1, 2]
# Generate all combinations of elements 0,1 and 2 length 11
combinations = list(product(elements, repeat=11))
arr=[]
# extract all arrays(reduced) with 12 zero, 9 one, 6 two
for combo in combinations:
    if check_array_11(list(combo)):
      a=list(combo).copy()
      b=a.copy()
      #arranging based on index in 11 element array
      b=[b[-3]]+b[0:6]+[b[-2]]+b[6:8]+[b[-1]]
      c=b.copy()
      arr.append(c)
# print('total possible elements that has twelve 0s, nine 1s,
      six 2s ', len(arr))
# creating matrix H
H=np.zeros((11,11),dtype = 'complex_')
rotations={1:[[0,0,0]],
2:[[0,0,1],[0,1,0],[1,0,0]],
3:[[0,0,2],[0,2,0],[2,0,0]],
4:[[0,1,1],[1,0,1],[1,1,0]],
5:[[0,1,2],[1,2,0],[2,0,1]],
6:[[0,2,1],[1,0,2],[2,1,0]],
7:[[0,2,2],[2,0,2],[2,2,0]],
8:[[1,1,1]],
9:[[1,1,2],[1,2,1],[2,1,1]],
```

```
10:[[1,2,2],[2,1,2],[2,2,1]],
11:[[2,2,2]]}
# value of w = (1 + i x root(3))/2
w=(-1+1.7320508075688772j)/2
for i in range(len(H)):
  for j in range(len(H)):
    r=rotations[j+1][0]
    cell=0
    for k in range(len(rotations[i+1])):
      Or=rotations[i+1][k]
      dot=sum(l[0] * l[1] for l in zip(r, Or))%3
      cell+=w**dot
    H[i][j] = cell
# final array
f=[]
# Extract elements whose modulus of product with matrix H is root 27
for i in arr:
  b=i.copy()
  d=[w**j for j in b]
  e=[round(abs(i),3) for i in d@H]
  if False not in [i==5.196 for i in e]:
    f.append(b)
print('no of arrays whos product gives root 27: ',end='')
print(len(f))
print('sample elements from array: ')
print(f[10:20])
# creating list of array f+1, f+2, 2f, 2f+1, 2f+2
f_plus_1=[]
f_plus_2=[]
twof=[]
twof_plus_1=[]
twof_plus_2=[]
for i in f:
  j=i.copy()
  j1=[(k+1)%3 for k in j]
  f_plus_1.append(j1)
  j2=[(k+2)%3 for k in j]
  f_plus_2.append(j2)
  j3=[(2*k)%3 for k in j]
  twof.append(j3)
  j4=[(2*k+1)%3 for k in j]
  twof_plus_1.append(j4)
  j5=[(2*k+2)%3 for k in j]    twof_plus_2.append(j5)
```

Appendix G: Algorithm computation of Nonlinearity of Ternary S-boxes of length 3^n

```
import numpy as np
import math
# create the matrix of degree wanted
def create_matrix(n,v3):
  n = int(math.log(n,3))
  array_n=np.full((3**(n-1), 3**(n-1)), 3)
  for i in range(1,n):
    array_n=np.full((3**(i), 3**(i)), 3)
    row1 = np.hstack((v3, v3, v3))
    row2 = np.hstack((v3, np.mod(v3+1,array_n), np.mod(v3+2,array_n)))
    row3 = np.hstack((v3, np.mod(v3+2,array_n), np.mod(v3+1,array_n)))
    v3 = np.vstack((row1, row2, row3))
  return v3
# to transform matrix v to v_dash
def transform_matrix(v):
  n=len(v)
  for i in range(n):
    for j in range(n):
      if v[i][j]==1:
        v[i][j]=2
      elif v[i][j]==2:
        v[i][j]=1
  return v
r3 = math.sqrt(3)
# to transform matrix v_dash to complex form
def transform_matrix_2(v):
  v_=v.copy()
  v_=v.astype(complex)
  n=len(v)
  for i in range(n):
    for j in range(n):
      if v_[i][j]==0:
        v_[i][j]=1
      elif v_[i][j]==1:
        v_[i][j]=(-1+r3*1j)/2
      elif v_[i][j]==2:
        v_[i][j]=(-1-r3*1j)/2
  return v_
def transform_f(f):
  for i in range(len(f)):
```

```
    if f[i]==0:
      f[i]=1
    elif f[i]==1:
      f[i]=(-1+r3*1j)/2
    elif f[i]==2:
      f[i]=(-1-r3*1j)/2
  return f
def find_nf(f,v):
  if len(f)!=len(v):
    print('wrong dimention for matrix v and array f')
    return
  ln=len(f)
  res=f@v
  res=[abs(j) for j in res]
  res=max(res)
  nf=ln - res
  return nf
def f_to_fs(f):
  n=int(math.log(len(f),3))
  fs=[]
  f=[(n-len(str(i)))*'0'+str(i) for i in f]
  for i in range(n):
    temp=[]
    for j in range(len(f)):
      temp.append(int(f[j][i]))
    fs.append(temp.copy())
  return fs
def nf_(f):
  ans=[]
  n=len(f)
  f_s=f_to_fs(f)
  v3=np.array([[0, 0, 0],
               [0, 1, 2],
               [0, 2, 1]])
  vn=create_matrix(n,v3)
  vn=transform_matrix(vn)
  vn=transform_matrix_2(vn)
  for matrix in f_s:
    matrix_transformed=transform_f(matrix)
    nf = find_nf(matrix_transformed,vn)
    ans.append(nf)
  print(ans)
  print("\nThis is the minimum nf: ",round(min(ans),3))
```

```
# give s_box
# for number starting with 0. ignore the starting zeros.
# for 010 give 10. for 000 give 0. for 001 give 1
f=[]
nf_(f)
```

Publications in Journals and Presentations

Publications:

1. P Aboobacker and M Viji, "New Design of S-box based on Galois field of Odd Characteristic and analysis of its cryptographic attributes," *Advances and applications of Mathematical Sciences,* vol. 21, no. 10, August 2022, pp. 5935-5946.

2. Aboobacker Parammel and Viji Maniyil, "On Resilient Quaternary functions", *AIP Conference Proceedings,* 2829, pp. 05004-1 05004-8, July. 2023.

3. Aboobacker P and Viji M, "Noval Approaches For Spectral analysis of Rotation Symmetric Functions On MVL and Extraction Of Rotation Symmetric Ternary Bent Functions With Reduced Computational Complexity," (Communicated)

Presentations:

1. Aboobacker P, *A review on Substitution Permutation Networks in Information Communication System*, MESMAC, International Seminar, MES College, Mampad, Kerala, India, held during 15-16 January 2019.

2. Aboobacker P, *Prerequisites for Cryptographic Primitives on Many-valued Logic* , Instructional Workshop on Advances in Cryptography(IWAC-2023), St.Thomas College, Thrissur, Kerala, held during 23 -26 May 2023.

Bibliography

[1] Data Encryption Standard (DES). (1993). National Institute of Standards and Technology, Federal information processing standards publication 46-2: http://www.itl.nist.gov/fipspubs/fip46-2.htm.

[2] Daemen, J., Rijmen, V. (2020). The design of Rijndael. In Information security and cryptography. https://doi.org/10.1007/978-3-662-60769-5

[3] Shannon, C. E. (1949). Communication Theory of Secrecy Systems. Bell System Technical Journal, 28(4), 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

[4] Advanced Encryption Standard (AES). (2001). National Institute of Standards and Technology, Federal information processing standards publication 197. http://csrc.nist.gov/publications/fips/fips197/fips197.

[5] Epstain, G. (1993). Multiple valued logic design:an introduction/G.Epstain.-Boca Raton:CRC Press,,370.

[6] Bykovsky, A. Yu. (2021). Multiple-Valued Logic and Neural Network in the Position-Based Cryptography Scheme. Journal of Russian Laser Research, 42(5), 618–630. https://doi.org/10.1007/s10946-021-10000-7

[7] Astola, J., Stankovic, R.S. (2006). Signal Processing Algorithms and Multiple-Valued Logic Design Methods. 36th International Symposium on Multiple-Valued Logic (ISMVL'06), 16-16.

[8] Dubrova, E. (1999). Multiple-Valued Logic in VLSI: Challenges and Opportunities, Proceedings of NORCHIP'99.

[9] Zhenxian, F., Ying, L. (1995), Ternary error correcting codes, Journal of Electronics and Information Technology 17 , 182-186.

[10] Sokolov, A., Kazakova, N., Kuzmenko, L., Mahomedova, M. (2021). Prerequisites for Developing a Methodology for Estimating and Increasing Cryptographic Strength based on Many-Valued Logic Functions. CPITS I.

[11] Marella,Surya Parisa,Hemanth.(2020).Introduction to Quantum computing 10.5772/intechopen.94103.

[12] Shor, P.W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science, 124-134.

[13] https://www.nist.gov/programs-projects/post-quantum-cryptography.

[14] Sokolov A.V., Zhdanov O.N.(2018). Prospects for the Application of Many- valued Logic Functions in Cryptography, International Conference on Theory and Applications of Fuzzy Systems and Soft Computing, P. 331-339.

[15] Webster, A.F., Tavares, S.E. (1986). On the Design of S-Boxes. In: Williams, H.C. (eds) Advances in Cryptology — CRYPTO '85 Proceedings. CRYPTO 1985. Lecture Notes in Computer Science, vol 218. Springer, Berlin, Heidelberg.

[16] Sokolov, A., Zhdanov, O.N. (2019). Avalanche Characteristics of Cryptographic Functions of Ternary Logic. Radio Electronics, Computer Science, Control, 177-185.

[17] Sokolov, A., Zhdanov, O.N. (2019). Strict Avalanche Criterion of Four-Valued Functions as the Quality Characteristic of Cryptographic Algorithms Strenght. Siberian Journal of Science and Technology. 20(2) , 183-190.

[18] Sokolov, A., Radush, V. (2019). Avalanche Characteristics Of Nyberg Construction S-boxes Represented by the Many-valued Logic Functions, Informatics and Mathematical Methods in Simulation. 9. 111-119.

[19] Tokareva, N.N. (2015). Bent Functions: Results and Applications to Cryptography. 1-202.

[20] Trakhtman, A.M., Trakhtman, V.A.(1975). Elements of theory of discrete signals on finite intervals, Moscow: Sov. Radio.

[21] Kazakova.N, Sokolov,A.V.(2020). Spectral and Nonlinear Properties of Complete Quaternary Code. Conference: Cybersecurity Providing in Information and Telecommunication Systems.

[22] Sokolov, A. and Noel, D.T.V. (2019). Nonlinear Properties of Rijndael S-boxes Represented by the Many-Valued Logic Functions. In CybHyg,96-106.

[23] Kazakova, N., Karpinski, M.P., Sokolov, A., Gancarczyk, T. (2021). Nonlinearity of Many-Valued Logic Component Functions of Modern Cryptographic Algorithms S-boxes. Procedia Computer Science. 192, 2731-2741.

[24] Rothaus, O.S. (1976). On "Bent" Functions. J. Comb. Theory, Ser. A, 20, 300-305.

[25] Dobbertin, H. (1995). Construction of bent functions and balanced Boolean functions with high nonlinearity. In: Preneel, B. (eds) Fast Software Encryption. FSE 1994. Lecture Notes in Computer Science, vol 1008. Springer, Berlin, Heidelberg.

[26] Tang, C., Xiang, C., Qi, Y., Feng, K. (2017). Complete Characterization of Generalized Bent and 2k-Bent Boolean Functions. IEEE Transactions on Information Theory, 63, 4668-4674.

[27] Paterson,K.G.(2001). Sequences For OFDM and Multi-code CDMA: two problems in algebraic coding theory, Sequences and their applications. Seta 2001. Second Int. Conference. Proc. Berlin: Springer, 46–71..

[28] Sokolov, A. V. (2013). Constructive method for the synthesis of nonlinear S-boxes satisfying the strict avalanche criterion. Radio Electronics and Communications Systems, 56(8), 415–423.

[29] Mazurkov, M.I., Barabanov, N.A., Sokolov, A. (2013). The key sequences generator based on bent functions dual couples,Odes'kyi Politechnichnyi Universytet. Pratsi 3(42), 150-156.

[30] Khymenko, M.V. Sokolov, A.V. (2022). Improvement of the pseudorandom key sequences generation algorithm based on cellular automaton and many-valued logic bent-sequences. Informatics and mathematical methods in simulation. 12. 137-143.

[31] Sokolov,A Zhdanov,O.N Barabanov,N.A(2016) Pseudo-random key sequence generator based on triple sets of bent-functions, PFMT, 1 (26), 85-91.

[32] Sokolov, A.V Zhdanov, O.N (2016). Regular synthesis method of a complete class of ternary bent-sequences and their nonlinear properties. Journal of Telecommunication, Electronic and Computer Engineering, 8, 39-43.

[33] Sokolov, A. V. (2020). Synthesis Method of Ternary Bent-Functions of Three Variables. Radio Electronics, Computer Science, Control, 82–89.

[34] Siegenthaler, T (1984). Correlation-immunity of nonlinear combining functions for cryptographic applications (Corresp.). IEEE Transactions on Information Theory, 30(5), 776–780

[35] Siegenthaler,T(1985)Decrypting a class of stream ciphers using ciphertext only,IEEE Trans. Computers,C-31(1),81–84.

[36] K.Gopalakrishnan,K. D.R Stinson,D.R(1995). Three characterizations of non-binary correlation-immune and resilient functions, Designs, Codes andCryptography, 5, 241-251).

[37] Feng,D(1999). Three characterizations of correlation-immune functions over rings Z_n, Theoretical Computer Science, 226, 37–43.

[38] Sokolov,A.V Zhdanov,O.N(2020).Correlation immunity of three-valued logic functions, Journal of Discrete Mathematical Sciences and Cryptography, 25(6), 1649-1665.

[39] Bakunina, E., Dykyi, O. (2022). Synthesis method for S-boxes satisfying the criterion of correlation immunity of Boolean and 4-functions. Journal of Discrete Mathematical Sciences and Cryptography,1-13.

[40] Kazakova, N., Sokolov, A., Troyanskiy, A. (2021). Correlation Immunity of Many-Valued Logic Component Functions of Modern Cryptographic Algorithm S-Boxes, II International Scientific and Practical Conference ≪Intellectual Systems and Information Technologies≫At: Odessa, Ukraine(2022).

[41] Zhdanov,O.N. Sokolov,A.V(2015) Algorithm of construction of optimal according to the criterion of zero correlation nonbinary S-boxes,Problems of physics, mathematics, and technics, 3 (24), 94-97.

[42] Pieprzyk, J., Qu, C. (1999). Fast Hashing and Rotation-Symmetric Functions. , Journal of Universe Computer.Science(5),20-31.

[43] Stănică, P., Maitra, S. (2003). Rotation symmetric Boolean functions - Count and cryptographic properties,Electronic Notes in Discrete Mathematics , 15, 139-145.

[44] Stănică, P., Maitra, S. (2003). A constructive count of rotation symmetric functions.Information Processing Letters , 88, 299-304.

[45] Li, Y. (2008). Results on rotation symmetric polynomials over GF(p).Information Sciences: an International Journal , 178, 280-286.

[46] Stănică, P., Maitra, S., Clark, J.A (2004). Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. Lecture notes in computer Science,Springer Verlag, 3017, 161-177

[47] Moraga,C(2008).Permutations under Spectral Transforms, 38th International Symposium on Multiple Valued Logic (ismvl 2008), Dallas, TX, USA, 76-81

[48] Sarkar, S., Maitra, S. (2007). Construction of Rotation Symmetric Boolean Functions on Odd Number of Variables with Maximum Algebraic Immunity. In: Boztaş, S., Lu, HF.(. (eds) Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 2007. Lecture Notes in Computer Science, vol 4851. Springer, Berlin, Heidelberg.

[49] Moraga, C., Stankovic, R., Astola, J. (2018). On the Reed-Muller-Fourier Spectrum of Multiple-Valued Rotation Symmetric Functions. IEEE Xplore.241-246. https://doi.org/10.1109/ISMVL.2018.00049.

[50] Moraga, C., Stankovic, M., and Stankovic, R(2020). On ternary symmetric bent functions.2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL), 76-81.

[51] Dey, S., Ghosh, R. (2018). 4, 8, 32, 64 Bit Substitution Box Generation Using Irreducible or Reducible Polynomials Over Galois Field GF(Pq) for Smart Applications. Lecture Notes in Intelligent Transportation and Infrastructure.

[52] Arshad, B., Siddiqui, N., Hussain, Z., Ehatisham-ul-Haq, M. (2022). A Novel Scheme for Designing Secure Substitution Boxes (S-Boxes) Based on Mobius Group and Finite Field. Wireless Personal Communications, 124, 3527 - 3548.

[53] Sokolov, A. V. (2013). Constructive method for the synthesis of nonlinear S-boxes satisfying the strict avalanche criterion. Radio Electronics and Communications Systems, 56(8), 415–423.

[54] Zhdanov,O.N, Sokolov,A.V(2015) Algorithm of construction of optimal according to the criterion of zero correlation nonbinary S-boxes, Problems of Physics, Mathematics, and technics, 3 (24), 94-97 .

[55] Zhdanov, O.N., Sokolov, A. (2016).Block symmetric cryptographic algorithm based on principles of variable block length and many-valued logic, Far East Journal of Electronics and Communications, 16, 573-589.

[56] Sokolov, A., Zhdanov, O.N. (2017). Nonlinear Nyberg Construction Transforms Over Isomorphic Representations Of Field Galois, System analysis and applied information science». (3) 59-67.

[57] Khan, M., Munir, N. (2019). A Novel Image Encryption Technique Based on Generalized Advanced Encryption Standard Based on Field of Any Characteristic. Wireless Personal Communications, 109(2), 849–867.

[58] Lidl, R., and Niederreiter, H. (1996). Polynomials over Finite Fields. In Finite Fields (Encyclopedia of Mathematics and its Applications, pp. 83-146). Cambridge: Cambridge University Press.

[59] Cid, C. (2006). Algebraic Aspects of the Advanced Encryption Standard. In Springer eBooks.

[60] Réjane Forrié. (1990). The Strict Avalanche Criterion: Spectral Properties of Boolean Functions and an Extended Definition. Lecture Notes in Computer Science, 450–468.

[61] Logachev, O.A., Salnikov, A.A., Yaschenko, V. (2012). Boolean Functions in Coding Theory and Cryptography, MCNMO, Moscow

[62] Sokolov A.V., Overchuk Yu.S.(2018). On the Possibility of Synthesizing the Algebraic Normal Form of Quaternary Functions Over a Field GF(4), The First International Scientific-Practical Conference on Problems of Cybersecurity of Information and Telecommunication Systems, 384-388.

[63] Mazurkov, M.I., Sokolov, A., Barabanov, N.A. (2016). Synthesis method for bent sequences in the Vilenkin-Chrestenson basis. Radioelectronics and Communications Systems, 59, 510-517.

[64] Carlet, C., Crama, Y., Hammer, P.L (2010). Boolean Functions for Cryptography and Error-Correcting Codes, Boolean Methods And Models in Mathematics, Computer Science and Engineering, 257-397 (2010).

[65] Logachev, O.A., Salnikov, A.A., Yaschenko, V. (2012). Boolean Functions in Coding Theory and Cryptography., Universities Press (India) Private Limited, ,334 (2017).

[66] Camion, P., Carlet, C., Charpin,P., Sendrier, N. (2007). On Correlation-immune functions,CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology Springer EBooks, 86–100.

[67] Sokolov, A. V., Zhdanov, O. N. (2018). The class of perfect ternary arrays, System Analysis and Applied Information Science, 0(2), 47–54.

[68] Aboobacker,P., Viji, M.(2023). On quaternary Resilient functions,2829(1). AIp Conference Proceedings,05004-1-05004-8.

[69] Stancovic,M.M., Morga.C, Stankovic,R.S(2017) Some spectral invariant operations for multiple-valued functions with homogeneous disjoint products in the polynomial form, Proceedings of 47 the Int. symp.Multiple-valued logic, Federiction NB,Canada.IEEE Press, 61-66.

[70] Nyberg,K.(1994) Differentially uniform mappings for cryptography, Ad-vances in cryptology. Proc. of EUROCRYPT'93, Lecture Notes in Computer Science.765, 55.

[71] Liu,J., Tong,X., Zhang,M., Wang,Z(2020). The Design of S-box Based on Combined Chaotic Map, 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Shenzhen, China, 350-353 .