Cryptographic Functions and Encryption Schemes for Images and 3D Objects

Esam Elsheh

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

2011

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Esam Elsheh

Entitled: Cryptographic Functions and Encryption Schemes for Images and 3D Objects

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical & Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. F. Haghighat

_____ External Examiner
Dr. F. Benkhaldoun

_____ External to Program
Dr. L. Kadem

_____ Examiner
Dr. D. Qiu

_____ Examiner
Dr. M. Kabir

_____ Thesis Co-Supervisor
Dr. A. Ben Hamza

_____ Thesis Co-Supervisor
Dr. A. Youssef

Approved by _____
Dr. M. Kahirizi, Graduate Program Director

_____
Dr. Robin A.L. Drew, Dean
Faculty of Engineering & Computer Science

ABSTRACT

Cryptographic functions and encryption schemes for images and 3D objects

Esam Elsheh, Ph.D.

Concordia University, 2011

One great challenge in data security is to design computationally efficient cryptographic algorithms and protocols that are necessary to keep systems secure, particularly when communicating through untrusted networks such as the Internet. Cryptography is a hierarchical science that can be divided into several sub-layers. At the highest layer, cryptographic protocols are used to provide security in various applications such as online banking, remote login and secure e-mail. These protocols rely on cryptographic algorithms to achieve their required security objectives. At the lowest level of this hierarchy, we have the components that are used to build these cryptographic algorithms.

The first problem addressed in this thesis is the design of cryptographic functions which are the basic building blocks of symmetric key primitives. We introduce a new measure for the cryptographic functions strength called *nonlinearity-profile*. We identify the existence of the linear structure in the class of rotation symmetric Boolean functions. Motivated by the better cryptographic bounds that can be achieved by functions over $GF(p)$, $p > 2$, we extend various cryptographic ideas from $GF(2)$ to the $GF(p)$ case.

Several attempts were made to construct public key cryptosystems based on trapdoor Boolean permutations that can be implemented efficiently. In this thesis, we analyze the security of one of the proposed cryptosystem based on Boolean permutations. In particular, we show that the construction suggested by Wu and Varadharajan is insecure by presenting an efficient cryptanalytic attack that allows the cryptanalyst to invert this class of Boolean permutations without the knowledge of the secret key parameters.

The rest of the thesis is devoted to the analysis and design of various cryptographic algorithms for image processing and 3D graphics. We show that image encryption schemes

based on parameterized discrete transforms represent typical examples of insecure ciphers. All the building blocks of these schemes are linear, and hence, breaking these scheme using a known plaintext attack is equivalent to solving a set of linear equations. We also invalidate the argument of relying on the visual quality of the encrypted image ciphertext by providing an experimental result for a trivially insecure system that produces ciphertext images with the same property.

In various applications, such as military documents and sensitive business data, the necessity of keeping these documents secret and safe must be firmly assured. In recent years, digital multimedia elements such as 2D images and 3D models have been considered as important as any other text sensitive information, and several image protection techniques have been proposed to assure the security of secret images. We show that the matrix-based secret image sharing scheme proposed by M. Rey is not ideal for practical usage due to the fact any participant in the scheme can recover the original image without the need to combine his/her share with any other participant. We also propose a geometric framework for 3D secret sharing. To decrease the amount of the information that must be kept secret, we use two lossless data compression algorithms, Huffman Coding and ZLIB, prior to splitting the 3D models. The experimental results on several 3D models indicate the feasibility of the proposed approaches.

ACKNOWLEDGMENTS

I would like to thank my supervisors, Dr. Amr Youssef and Dr. Abdessamad Ben Hamza, for the patient guidance, encouragement and advice they have provided throughout my time as their student. I have been extremely lucky to have supervisors who cared so much about my work, and who responded to my questions and queries so promptly.

I would like to express my deepest gratitude for the constant support, understanding and love that I received from my mother, my wife Ghada, and my family during the past years.

Completing this work would have been all the more difficult were it not for the support and friendship provided by the other members of the Crypto Lab in Concordia. I am indebted to them for their help.

Finally, I am very grateful for the funding support of the Libyan Ministry of Education and Scientific Research for making this research possible.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Framework and Motivation

Recent advances in computer technology have contributed to the emergence of data protection and the need to communicate information without the fear of being intercepted by unintended recipients. To achieve this objective, the information is encoded in such a way that the intended receiver is the only one who can retrieve its contents. This process has developed into the science of cryptography. The encoding technique has been termed encryption, which is the transformation of the original data into a form that is unreadable without the appropriate secret knowledge. The development of cryptography has been paralleled by the development of cryptanalysis, the art and science of how to compromise such cryptographic systems.

This chapter contains problems definition, motivation, objectives, and a brief review of essential concepts and definitions which we will refer to throughout the thesis. We also present a short summary of material relevant to cryptography and multimedia security.

1.1 Problems definition

Cryptography [1] is a hierarchical science that may be divided into several sub-layers. At the highest layer, cryptographic protocols are used to provide security in various applications such as online banking, remote login and secure e-mail. These protocols rely on cryptographic algorithms to convey these messages securely. In other words, cryptographic algorithms can be considered as the core for these cryptographic protocols. These algorithms perform the mathematical transformations to provide data protection. Examples for these algorithms are the RSA [2] public key system and the Advanced Encryption (AES) [3] symmetric key algorithm. At the lowest level of this hierarchy, we have the components that are used to build these cryptographic algorithms. In the case of symmetric key algorithms such as block ciphers, stream ciphers and hash functions, Boolean functions and S-boxes represent these primitive components. Since the security of the cryptographic algorithms depend on these Boolean functions building blocks, one can argue that the overall system security also depends on these primitives. In this thesis, we systematically study various aspects of this hierarchy starting from the primitive components to the cryptographic applications for delivering a secure text and multimedia contents.

1.1.1 Properties of cryptographic functions

Symmetric key primitives, including block cipher, stream ciphers and hash functions, are used to secure many applications such as mobile phones, smart cards, Internet and wireless communications, and multimedia applications. Boolean functions are the basic building blocks of these cryptosystems. In some applications, these functions have only a single output. In other applications, these Boolean functions have multiple outputs. Multiple

output Boolean functions are called *vectorial Boolean functions* or *Substitution Boxes* (S-boxes) [4]. The study of different cryptographic properties of Boolean functions is important because of the strong connections between known cryptanalytic attacks and these properties. In many cases, the ciphers security against a given attack can be assured by the existence of a specific property in the underlying Boolean functions or S-boxes. In this thesis, we introduce a new concept to evaluate the strength of the cryptographic Boolean function called *nonlinearity-profile*. This measure calculates the function nonlinearity when some of the function coordinates are fixed.

Rotation Symmetric Boolean functions (RSBFs) [5,6] have been extensively studied in recent years because of their small search space and their richness in terms of cryptographic properties. However, there are some other characteristics these functions might have which should be avoided. Namely, the existence of non trivial (i.e., non-zero) linear structure in Boolean functions is regarded as a weakness and a function possessing this property is considered vulnerable to cryptanalysis. Hence, in this thesis we examine the existence of linear structures in RSBFs.

The existence of a tradeoff between the cryptographic properties in $GF(2)$ functions has potential consequences on the security of a cryptosystem. For example, we cannot construct a function over $GF(2)$ with the maximum order of correlation immunity and algebraic degree higher than 1. On the other hand, when the function is defined over $GF(p)$, $p > 2$, it is possible to construct a non-linear function that has a maximum order of correlation immunity. Thus, motivated by the better cryptographic properties bounds that these functions can achieve, we extend various cryptographic properties from $GF(2)$ to prime finite field $GF(p)$.

1.1.2 Public key cryptosystems based on Boolean permutations

Most widely adopted public key cryptosystems are based on two underlying hard problems, discrete logarithm-based such as Diffie-Hellman, and integer factorization-based such as the RSA algorithms [1]. Regardless of the considerable improvement that has been made in implementing these systems, they are still relatively computationally costly, particularly in resource-constrained environments. To tackle this problem, several attempts were made to construct public key cryptosystems based on trapdoor Boolean permutations that can be implemented efficiently. The main idea is to find a class of efficiently computable random-looking permutations with compact representations, which can be easily inverted given some trapdoor information.

1.1.3 Image encryption schemes based on discrete transforms

Discrete transforms, such as Fourier, Cosine, and Wavelets, have been widely applied to diverse fields including image processing. Motivated by the wide available spectrum of possible applications, the majority of these transforms have been generalized by adding extra parameters to their original mathematical definitions. A common characteristic of these generalized transforms is that they have a relatively larger number of independent parameters as compared to their corresponding original forms. This led many researchers to propose image encryption algorithms based on a single or multiple steps of these transforms where their parameters are used as encryption keys. A careful analysis of these algorithms reveals two major problems associated with their practical implementation, namely, the security and the performance in terms of both bandwidth and throughput.

1.1.4 Secret sharing schemes for images and 3D models

The continuing advancements in computer technologies and the rapid increase in internet users have led to the increasing usage of network-based data transmission. In various applications, such as military documents and sensitive business data, the necessity of keeping these documents available and confidential must be firmly implemented. In recent years, digital multimedia contents such as 2D images and 3D models have been considered as important as any other sensitive information. Therefore, several 2D image-protection techniques have been proposed to assure the security of 2D images. Most of these systems enhance availability by providing full replication. Client-side encryption can protect information confidentiality even when storage nodes are compromised. Secret sharing schemes offer an alternative to these approaches that provides both information confidentiality and availability. These schemes encode, replicate, and divide information into multiple pieces, or shares, that can be stored at different storage nodes. The system can only reconstruct the information when enough shares are available.

1.2 Objectives

Our objectives may be summarized as follows:

- Investigate various cryptographic properties for different classes of Boolean functions, and then generalize some of the previous results on cryptographic Boolean functions to functions defined over $GF(p)$, $p > 2$.

- Study the security of public key cryptosystems based on Boolean permutations. In particular, we show that the constructions suggested by C. Wu and V. Varadharajan are insecure by presenting an attack that can be used to invert this class

of Boolean permutations without the knowledge of the secret key parameters.

- Provide a comprehensive analysis on the security and efficiency of the image encryption schemes based on discrete multiple parameters transforms.

- Analyze the security of the matrix-based secret sharing scheme for images proposed by M. Rey and present two secret sharing approaches for 3D models.

1.3 Background

Throughout history, people have always felt the need to communicate information without the fear of being intercepted by unintended recipients. To achieve this objective, the information was encoded in such a way that the intended receiver is the only one who can retrieve its contents. This process has developed into the science of cryptography. The development of cryptography has been paralleled by the development of cryptanalysis, the art and science of how to compromise such cryptographic systems. The encoding technique has been termed *encryption*, which is the transformation of the original data (plaintext), into a form that is unreadable without the appropriate secret knowledge, referred to as a *key*. This unreadable information is called *ciphertext*. To get back the original contents, the intended receiver who has the key will perform a reverse operation, called *decryption*. A general architecture of encryption/decryption process is shown in Figure 1.1.

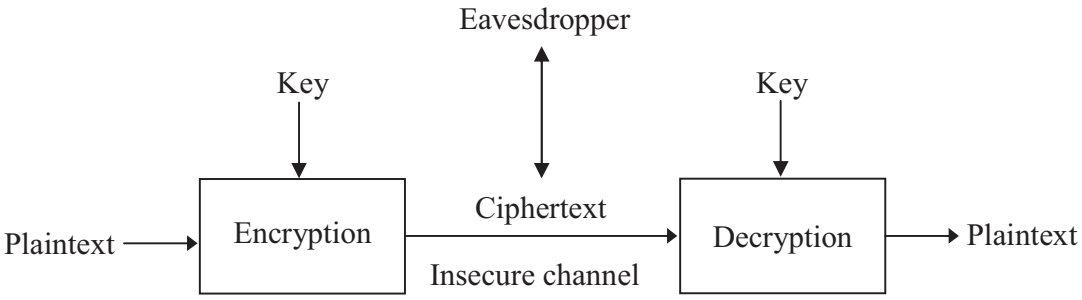


Figure 1.1: General architecture of a cryptosystem.

Cryptographic techniques are classified into two categories: symmetric-key and public-key. When using symmetric-key techniques, both the encryption and decryption keys have to be kept secret. On the other hand, when using public-key algorithms, one part of the key will be public (e.g., the encryption key or verification key) and the other key has to be kept private. Despite the key management problems associated with symmetric-key algorithms, they are invaluable components in any cryptographic system. This is mainly because they

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table 1.1: Example of truth table representation of a 3-variable Boolean function.

are orders of magnitudes faster than their public-key counterparts. Public-key primitives are usually constructed based on some number theoretic ideas (e.g., RSA [2] which is based on the difficulty of factoring large integers and ElGamal cryptosystem [7] which is based on the difficulty of solving the discrete log problem). On the other hand, symmetric-key primitives are constructed using Boolean functions interconnected in such a way that achieve the security requirements of these primitives (e.g., confusion and diffusion [8] in block ciphers).

1.3.1 Boolean Functions

Boolean functions are the basic components of symmetric key systems which are fundamental tools in the design of all types of digital security protocols (e.g., communications, financial and e-commerce). There are many ways to represent Boolean functions. In this thesis, we focus on the two relevant representations: Truth Table (TT) and the Algebraic Normal Form (ANF).

Truth Table Representation

We represent a Boolean function by the output column of its truth table as a binary string T_f of length 2^n as follows: $T_f = [f(00\cdots00)f(00\cdots01)\ldots f(11\cdots10)f(11\cdots11)]$. Table 1.1 shows an example for the truth table of a 3-variable function, where $T_f = [00110110]$.

A function f is said to be *balanced* if its output column in the truth table has an equal number of 0's and 1's, i.e. $wt(f) = 2^{n-1}$ where wt denotes the Hamming weight of the function f, i.e., the number of $1's$ in its truth table.

ANF Representation

Another representation of an n-variable Boolean function f, called the *Algebraic Normal Form* (ANF), is a multivariate polynomial given by

$$f(x_1, \cdots, x_n) = a_0 \bigoplus_{i=1}^{n} a_i x_i \bigoplus_{1 \le i < j \le n} a_{ij} x_i x_j \oplus \ldots \oplus a_{1,\ldots,n} x_1 \cdots x_n,$$

where the coefficients $a_0, a_1, a_{1,\ldots,n} \in \mathbb{F}_2$. The algebraic degree denoted by $deg(f)$ is the number of variables in the highest order term with a nonzero coefficient. The set of all n-variable Boolean functions is equal to 2^{2^n}. Enumerating all Boolean functions with $n \geq 6$ is beyond the computational capability of today's PCs.

1.3.2 Discrete transforms

The purpose of the transform coding is to decompose a set of correlated signal samples into a set of uncorrelated spectral coefficients with energy concentrated in as few coefficients as possible. The uneven distribution of signal energy in the frequency domain has made signal decomposition an important practical problem. Recently, discrete transforms have been widely used in image processing applications. Specifically, transform-based data encryption techniques have become attractive for many recent communication systems. In recent years, there has been an enormous interest in developing parameterized versions of the existing fixed transforms [9–13]. These parameterized transforms were proposed to be used in a wider range of applications compared to its original version to provide more flexibility in representing, interpreting and processing of signals. For example, the independent

parameters of a fractional discrete transform have been used as an additional secret key for watermarking [14] and for encryption [15–17]. One of the widely used discrete transforms is the Discrete Fourier Transform (DFT). This transform is invertible and has been extensively employed in signal processing and analysis.

The discrete Fourier transform pair is defined by

$$X(k) = \sum_{n=0}^{N-1} \mathbf{x}(n) e^{-j2\pi kn/N}, \quad k = 0, 1, 2, \ldots, N-1 \tag{1.1}$$

$$\mathbf{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}, \quad n = 0, 1, 2, \ldots, N-1 \tag{1.2}$$

A DFT matrix is an expression of a discrete Fourier transform as a matrix multiplication. If **F** is $N \times N$ discrete Fourier transform matrix, then the 1D-DFT of a $N \times 1$ vector $\mathbf{x}$ is:

$$\mathrm{DFT}(\mathbf{x}) = \mathbf{F} \cdot \mathbf{x}$$

and the 2D-DFT of a matrix **P** is given by:

$$\mathrm{DFT}(\mathbf{P}) = (\mathbf{F} \cdot (\mathbf{F} \cdot \mathbf{P})^T)^T$$

1.3.3 Secret sharing

Secret sharing is a procedure for sharing a secret among a number of participants such that only the qualified subsets of participants have the ability to reconstruct the secret. The simplest secret splitting method is the $(2, 2)$-scheme [1] where the secret K is split into two shares X and Y. Neither X nor Y independently provide any information about the secret. Let $K = k_1, \ldots, k_n$ be a binary string of length n, called the *secret*.

1. for $1 \leq i \leq n$, let $x_i \in \mathbb{F}_2$ be chosen at random.

2. for $1 \leq i \leq n$, let $y_i = x_i + k_i \bmod 2$.

3. then $X = x_1, \ldots, x_n$ and $Y = y_1, \ldots, y_n$ are two shares corresponding to the secret K.

4. To recover the secret, K is computed as $K = X + Y \bmod 2$.

The majority of existing secret sharing schemes are generalized within the so-called (t, n)-*threshold framework* [1]. This framework splits the content of a secret message into n shares in a way that requires the presence of at least t shares for the secret message reconstruction. If $t = n$, then all the shares are required in the (n, n)-threshold scheme to recover the secret. Conversely, the lost of any of the produced shares results in inaccessible secret messages. Therefore, apart from the simplest $(2, 2)$-schemes that are commonly used as a private key cryptosystem solution, the general (t, n)-threshold schemes with $t < n$ are often the point of interest due to their ability to recover the secret message even if several shares are lost. In this case, any possible combinations of t shares can be used to recover the secret message. Since protection against cryptanalytic attacks, including brute force enumeration, should remain unchanged regardless of how many shares are available until the threshold t is reached, the use of $(t - 1)$ shares should not reveal any valid information about the secret compared to that obtained by only one share.

Shamir (t, n)-Threshold Scheme [18]

Let $x_{1,}, \ldots, x_n$ be n distinct non-zero elements of $\mathbb{F}_P$. Given a secret $K \in \mathbb{F}_P$, the n shares are generated as follows:

1. Let $a_1, \ldots, a_{t-1}$be chosen independently at random from $\mathbb{F}_P$.

2. Define
$$a(x) = K + \sum_{j=1}^{t-1} a_j x^j \bmod p. \tag{1.3}$$

3. For $1 \leq i \leq n$, let the share $s_i = (x_i, y_i)$, where $y_i = a(x_i)$.

Given t shares, K can be computed using the Lagrange interpolation polynomial given by

$$K = \sum_{j=1}^{t} y_{i_j} \prod_{\substack{1 \leq m \leq t \\ m \neq j}} \frac{x_{i_m}}{x_{i_m} - x_{i_j}}. \tag{1.4}$$

1.4 Thesis overview

The remainder of the thesis is organized as follows:

- ❏ In Chapter 2, we investigate various cryptographic properties for different classes of Boolean functions, and then we generalize some of the previous results on cryptographic Boolean functions to functions defined over $GF(p)$.

- ❏ In Chapter 3, we analyze the security of the public key cryptosystem PCK2 family proposed by Wu and Varadharajan. In particular, we show that the suggested construction for the PCK2 family is insecure by presenting an efficient cryptanalytic attack that allows the cryptanalyst to invert the class of Boolean permutations used in the construction without the knowledge of the secret key parameters.

- ❏ In Chapter 4, we show that image encryption algorithms based on parameterized discrete transforms are inefficient and insecure for the practical applications usage.

- ❏ In Chapter 5, we provide an algebraic cryptanalysis of the matrix-based image secret sharing scheme. Also, we introduce a geometric framework for 3D secret sharing as a defence mechanism for 3D objects.

- ❏ In Chapter 6, we summarize the contributions in this thesis and propose some future research directions.

- ❏ In the bibliography section, we cite in references [90–97] the list of publications resulting from this thesis.

CHAPTER 2

Cryptographic Functions

Boolean functions play a major role in the construction of symmetric key primitives such as block ciphers, stream ciphers and hash functions. The security of these primitives depends on the cryptographic properties of the Boolean functions used in its construction. Various criteria, including balance, nonlinearity [19], resiliency [20] and algebraic immunity [21], have been proposed for measuring the cryptographic strength of Boolean functions. Let criterion C denote the cryptographic property of interest. We define the C-profile of the Boolean function as a measure that shows how this criterion degrades when we fix a subset of the input coordinates of the function. This is interesting from a cryptanalytic point of view, since fixing the coordinates of a cryptosystem is a well known cryptanalysis method. In Section 2.2, we introduce this concept and apply it to the nonlinearity property of the cryptographic Boolean function.

Due to its richness in terms of cryptographically properties along with its small search space ($2^{2^n/n}$ comparable to the whole space 2^{2^n}), the class of Rotation Symmetric Boolean functions (RSBFs) has become the main focus on searching for a Boolean function with good properties. Additionally, there are some other characteristics which may degrade the security of the underlying cryptographic primitives and should be avoided. For instance, a non trivial linear structure [22] other than all-zero is regarded as a weakness and the

function posses this characteristic is considered fragile and should not be used in cryptographic algorithms. Therefore, in Section 2.3 we examine the existence of linear structures in RSBFs.

Traditionally, cryptographic applications designed on hardware have always tried to take advantage of the simplicity of implementation functions over $GF(2)$ to reduce costs and improve performance. However, in recent years there has been growing interest of the implementation of cryptographic primitives based on $GF(p)$. We therefore, in Section 2.4, generalize some of the previous results on cryptographic Boolean functions to functions defined over $GF(p)$, $p > 2$. We first generalize Siegenthaler's construction to functions defined over finite field. Next, we characterize the linear structures of functions over $GF(p)$ in terms of their Walsh transform values. We then investigate the relation between the autocorrelation coefficients of functions over $GF(p)$ and their Walsh spectrum. We also derive an upper bound for the dimension of the linear space of the functions defined over $GF(p)$. Finally, we present a method to construct a bent function from semi-bent functions.

2.1 Algebraic preliminaries

In this section we introduce definitions and concepts used throughout this chapter.

Definition 2.1.1. The Walsh-Hadamard transform of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as [23]:

$$F(\mathbf{w}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus <\mathbf{w} \cdot \mathbf{x}>} \tag{2.1}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_n) \in \mathbb{F}_2^n$, and $< \mathbf{w} \cdot \mathbf{x} >$ denotes the inner product of $\mathbf{w}$ and $\mathbf{x}$, i.e., $< \mathbf{w} \cdot \mathbf{x} >= \oplus_{i=1}^{n} w_i x_i$.

A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called an *affine function* if $f(\mathbf{x})$ can be expressed as $f(\mathbf{x}) =$

$\boldsymbol{a} \cdot \mathbf{x} \oplus b$, where $\boldsymbol{a} \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2$. If $b = 0$ the function is called a *linear function*. The algebraic degree of the function $deg(f(\mathbf{x}))$, is the degree of the largest product term which exists in its Algebraic Normal Form (ANF_f). In this context, we use $\underset{d}{f}(\mathbf{x})$ to denote a Boolean function with algebraic degree equal d. The nonlinearity, NL, of f is the minimum distance between f and the set of affine functions. In terms of the Walsh transform, the nonlinearity of f is given by

$$NL(f) = 2^{n-1} - \frac{1}{2} \max_{\mathbf{w} \in \mathbb{F}_2^n} |F(\mathbf{w})| \tag{2.2}$$

For even n, bent functions [24] achieve the maximum nonlinearity $2^{n-1} - 2^{\frac{n}{2}-1}$. For odd n, a semi-bent function has nonlinearity $2(2^{n-2} - 2^{\frac{n-1}{2}-1})$. For odd $n > 7$, the maximum achievable nonlinearity is an open problem related to determining the covering radius of RM-codes [25].

Definition 2.1.2. [26] If two functions $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$, satisfy $g(\mathbf{x}) = f(\mathbf{x}A + \boldsymbol{b})$ with $\boldsymbol{b} \in \mathbb{F}_2^n$ and A is a nonsingular $n \times n$ matrix, we say that g is affinely equivalent to f.

Definition 2.1.3 (Linear Structure)**.** A vector $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ is called a *linear structure* of f if $f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \boldsymbol{\alpha})$ is constant. The $\mathbf{0}$ is a trivial linear structure for any Boolean function.

Definition 2.1.4 (Derivative of Boolean function)**.** The derivative of a Boolean function $f(\mathbf{x})$, with respect to a vector $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ is defined as:

$$d_{\boldsymbol{\alpha}} f(\mathbf{x}) = f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \boldsymbol{\alpha}) \tag{2.3}$$

If the functions has a linear structure $\boldsymbol{\alpha}$, then $deg(d_{\boldsymbol{\alpha}} f(\mathbf{x})) = 0$.

2.2 Nonlinearity profile of cryptographic Boolean functions

The security of symmetric key ciphers mostly depends on the cryptographic properties of their Boolean functions. In order for these cryptosystems to resist various statistical cryptanalytic attacks (e.g., correlation attack [27], differential cryptanalysis [28], linear cryptanalysis [29], and algebraic attacks [30]), the Boolean functions used in its construction have to satisfy these cryptographic properties.

In this section, we mainly investigate the nonlinearity-profile concept and its application to the nonlinearity-profile of Boolean functions. While nonlinearity measures the distance between the Boolean function and the set of affine functions, the nonlinearity profile measures the corresponding distance when some of the function coordinates are fixed. This is interesting from a cryptographic point of view, since fixing the coordinates of a cryptosystem is a well known cryptanalysis method. Similar notion has been previously considered by Carlet [31], where the hyper-bent function was introduced. This class of functions remains bent when any even integer k coordinate are fixed, $2 \leq k \leq n-2$.

2.2.1 Nonlinearity profile

The nonlinearity-profile depicts how the nonlinearity of the Boolean function degrades as we fix subsets of its input coordinates. Let NL denotes the nonlinearity of the function f. Then $NL(f|x_{i_1} \cdots x_{i_m})$ denotes the nonlinearity of the function obtained by fixing the input subset $(x_{i_1} \cdots x_{i_m}) \in \{x_1, \cdots, x_n\}$ to a specific value over $\mathbb{F}_2^m$. We define the average nonlinearity-profile of the function f as $NLP_m = E(NL(f|x_{i_1} \cdots x_{i_m}))$, $1 < m \leq n-2$, where the expectation $E(\cdot)$ is evaluated over $\{x_{i_1}, \cdots, x_{i_m}\} \in \mathbb{F}_2^m$. The minimum nonlinearity profile is defined as $NLP'_m = \min NL(f|x_{i_1} \cdots x_{i_m})$, $1 < m \leq n-2$, where

the minimum is evaluated over $\{x_{i_1}, \cdots, x_{i_m}\} \in \mathbb{F}_2^m$.

Example 2.2.1. Consider the two 5-variable Boolean functions $f = x_1x_3 \oplus x_2x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_5 \oplus x_2x_5 \oplus x_3x_5$ and $g = x_1x_3 \oplus x_3x_4 \oplus x_2x_5$. Both functions have $NL = 12$, correlation immunity $= 0$, algebraic degree $= 2$, absolute indicator [23] $= 32$ and posses $PC(1)$. Figure 2.1 shows how the average nonlinearity of both functions varies when fix subset of its input variables.

The minimum nonlinearity profile is also an important property for the cryptanalysts. Figure 2.2 shows that if we fix two inputs x_2x_3 of function g, the nonlinearity will collapse to zero, i.e., $NLP_2'(g) = 0$. Hence, an attacker can approximate the derived function by an affine function.

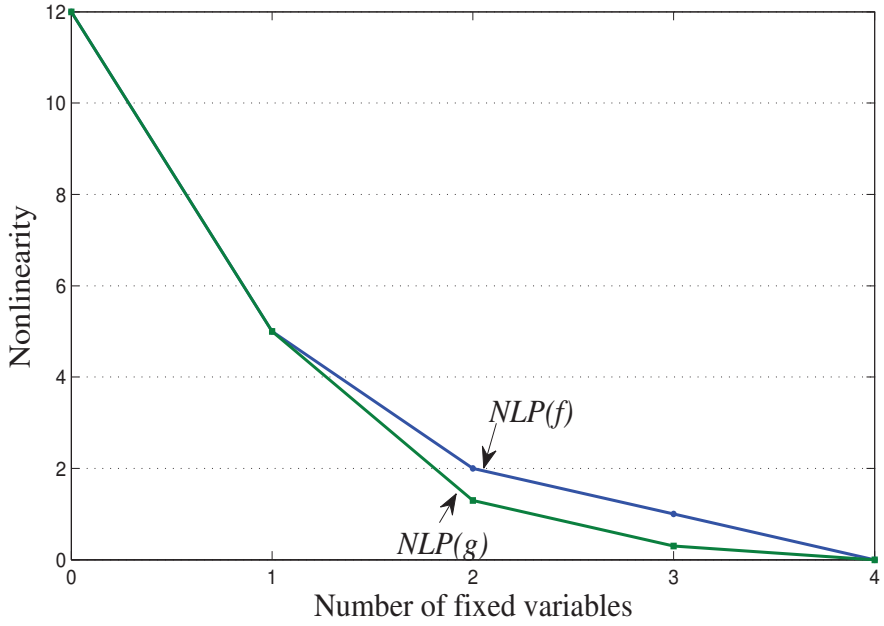


Figure 2.1: The nonlinearity profiles for f and g in Example 2.2.1.

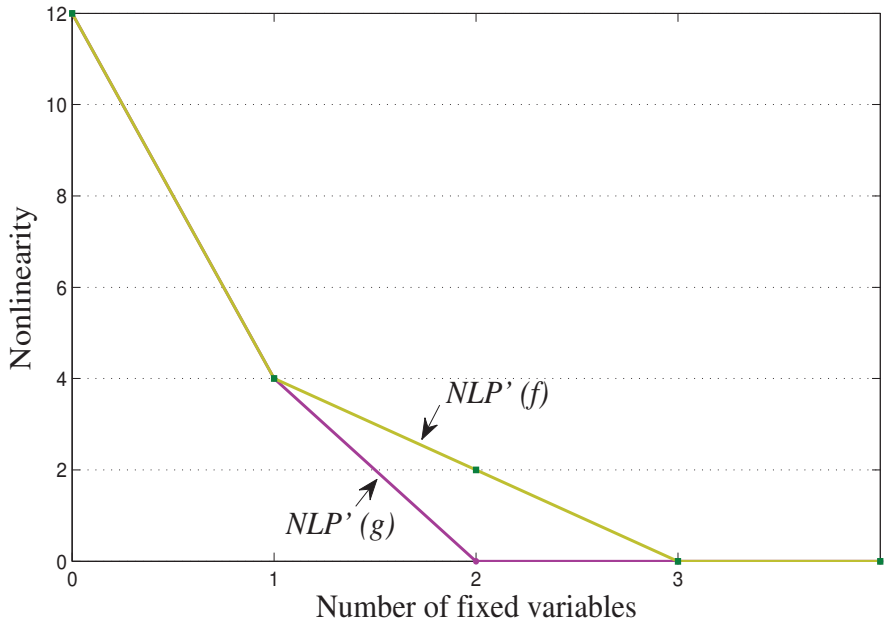


Figure 2.2: The minimum nonlinearity profiles for f and g in Example 2.2.1.

The nonlinearity-profile provides a good measure to differentiate between Boolean functions with the same nonlinearity. For example, using exhaustive search, one can show that for $n = 5$, balanced Boolean functions with maximum nonlinearity ($NL = 12$) can be classified into 144 classes based on their average nonlinearity profiles as shown in Figure 2.3.

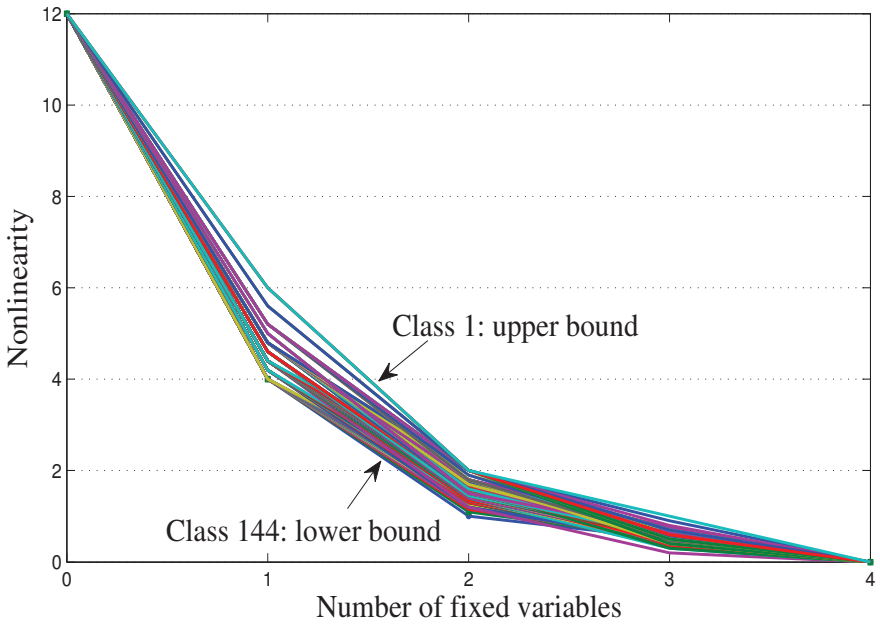


Figure 2.3: Nonlinearity profiles for all 5-variable balanced functions with NL =12 (144 Classes).

It should also be noted that affinely equivalent functions do not necessarily have the same nonlinearity profile.

Example 2.2.2. Let $f = x_1 \oplus x_1x_2 \oplus x_3 \oplus x_1x_4 \oplus x_3x_4 \oplus x_1x_2x_5 \oplus x_2x_3x_5 \oplus x_2x_4x_5$ and $g = x_1x_2 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_3x_4 \oplus x_2x_4x_5$. Although f and g are affinely equivalent, their nonlinearity-profiles (shown in Figure 2.4) are different.

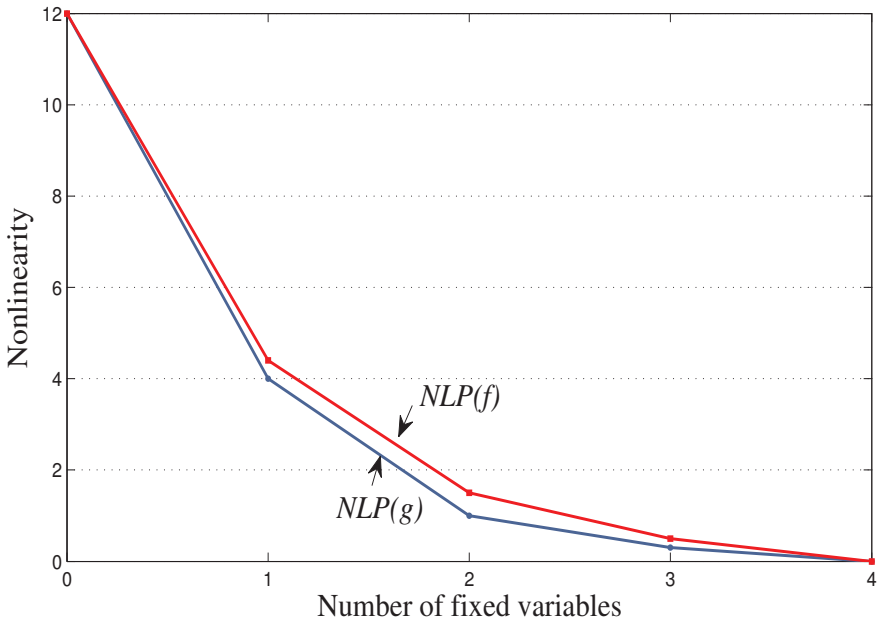


Figure 2.4: The nonlinearity profiles for f and g in Example 2.2.2.

It is important to note that adding any linear term to the function does not affect the nonlinearity profile, but it can change the balanedness of the function.

2.2.2 Graph representation of quadratic Boolean functions

A quadratic function can be represented by undirected graph with n nodes. The existence of a quadratic term $x_i x_j$ in the Boolean function is denoted by an arc in the graph connecting node i with node j. It is easy to show that Boolean functions corresponding to isomorphic graphs have the same nonlinearity profile. In [25], the quadratic bent functions, $n = 4$, are classified into 4 distinct graphs , depending on the Number of Quadratic Terms (NQT) that appear in their algebraic normal form as shown in Figure 2.5. It is important to note that, for $n = 4$ all the quadratic bent functions with the same NQT have the same nonlinearity-profile. While on the contrary, for $n > 4$ the quadratic functions with same NQT have

different nonlinearity-profiles.

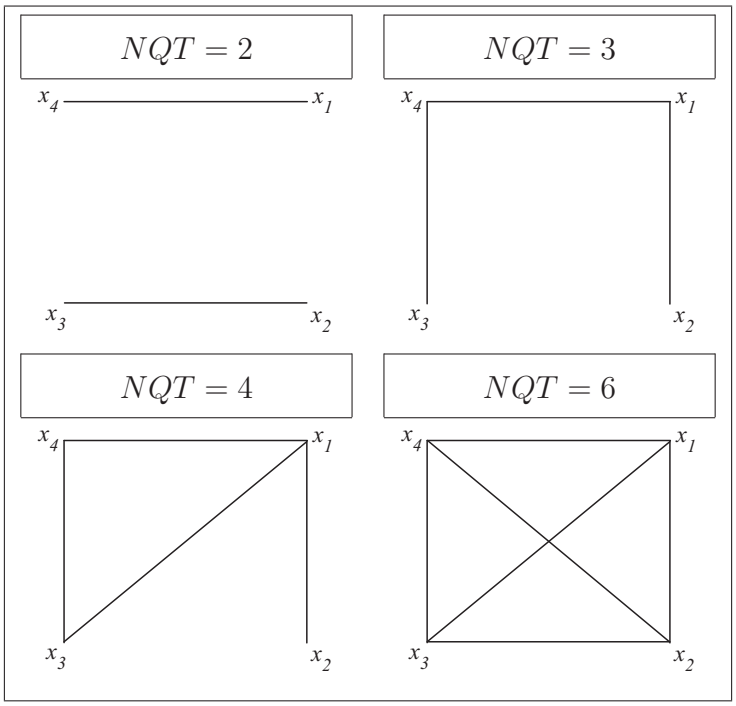


Figure 2.5: Graph representation for different classes of quadratic function $n = 4$, $NL = 6$.

Figures 2.6 and 2.7 show the graphs corresponding to different classes of quadratic functions with $NL = 12$ and $NL = 28$ (maximum achievable nonlinearity for $n = 5$ and $n = 6$, respectively). It should be noted that one can easily count all the quadratic functions corresponding to a given graph. For simplicity, we represent the graphs with its Degree Vector DV. By permuting the vector we get all the functions with the same nonlinearity profile. For example, when the number of quadratic terms equals 2, the degree vector $DV = [0\ 1\ 1\ 1\ 1]$, so the number of such functions $= \binom{5}{2}\binom{3}{2}/2! = 15$ functions. Similarly, for quadratic terms$= 3$: $DV = [2\ 1\ 1\ 1\ 1]$ the number of the isomorphic functions $= \binom{5}{2}\binom{3}{1} = 30$. Tables 2.1-2.3 show the average nonlinearity profile corresponding to the graphs for $n = 4, 5$ and 6.

NQT	NLP_1	NLP_2
2	2	0.33
3	2	0.50
4	2	0.66
6	**2**	**1**

Table 2.1: The best nonlinearity profiles for 4-variables quadratic functions.

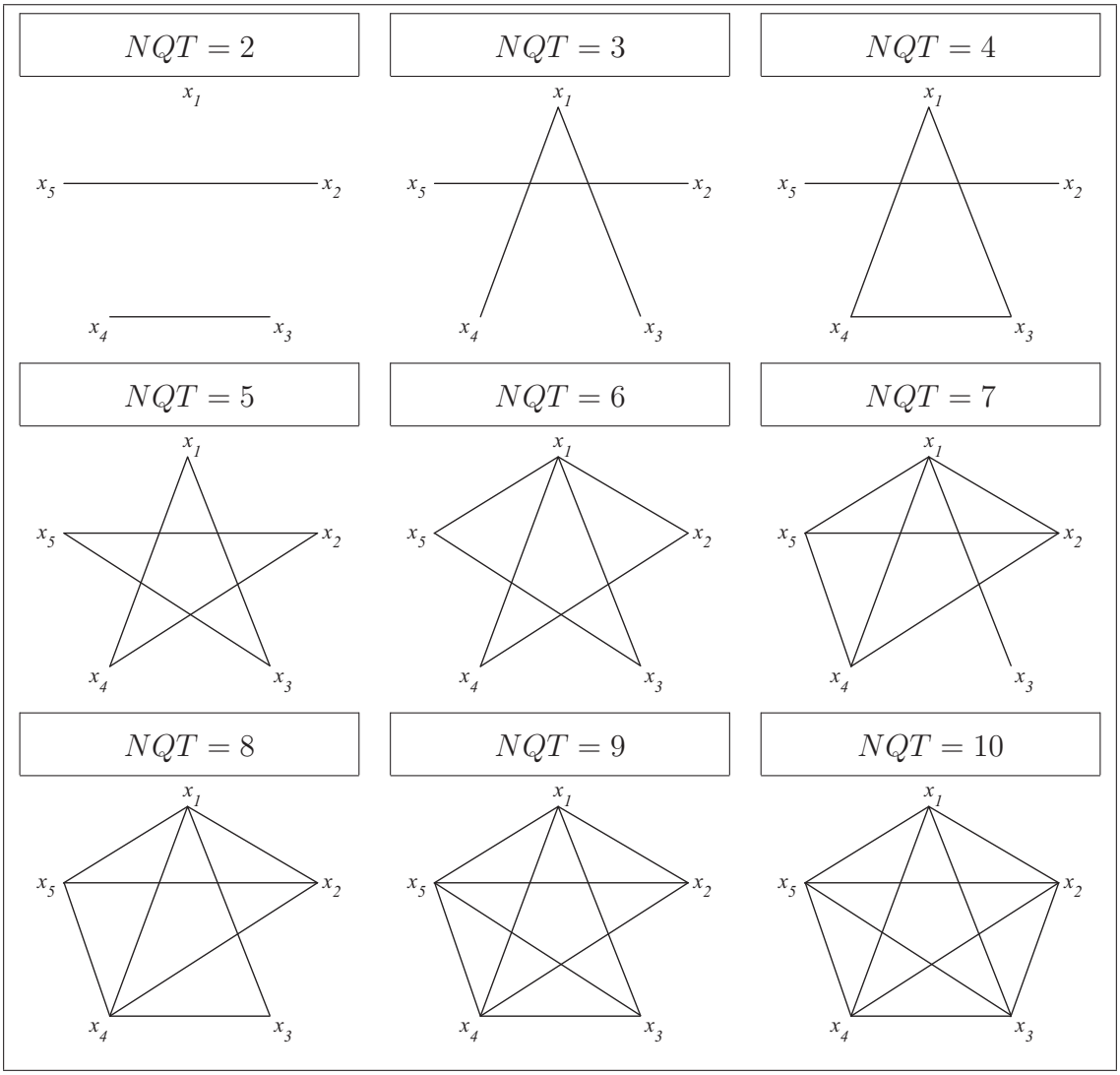


Figure 2.6: Graph representation of quadratic functions $n = 5, NL = 12$.

It is worth observing that some quadratic functions with specific number of quadratic terms do not achieve the maximum nonlinearity (e.g., for $n = 4$, $NQT = 5$ and $n = 6$,

NQT	NLP_1	NLP_2	NLP_3
2	4.4	1.2	0.2
3	4.8	1.6	0.3
4	5.2	2	0.4
5	6	2	0.5
6	6	2	0.6
7	5.6	2	0.7
8	5.2	2	0.8
9	4.8	2	0.9
10	**6**	**2**	**1**

Table 2.2: The best nonlinearity profiles for 5-variables quadratic functions.

NQT	NLP_1	NLP_2	NLP_3	NLP_4
3	12	4.4	1.2	0.2
4	12	4.53	1.4	0.26
5	12	4.8	1.6	0.33
6	12	5.06	1.8	0.4
7	12	5.46	2	0.46
8	12	5.6	2	0.53
9	12	5.73	2	0.6
10	12	5.33	2	0.66
11	12	5.46	2	0.73
12	12	5.06	2	0.8
13	12	5.2	2	0.86
15	**12**	**6**	**2**	**1**

Table 2.3: The best nonlinearity profiles for 6-variables quadratic functions.

$NQT = 14$). Based on our experimental results, it is clear that for $n \leq 7$, quadratic functions with all the quadratic terms have the optimum nonlinearity-profile.

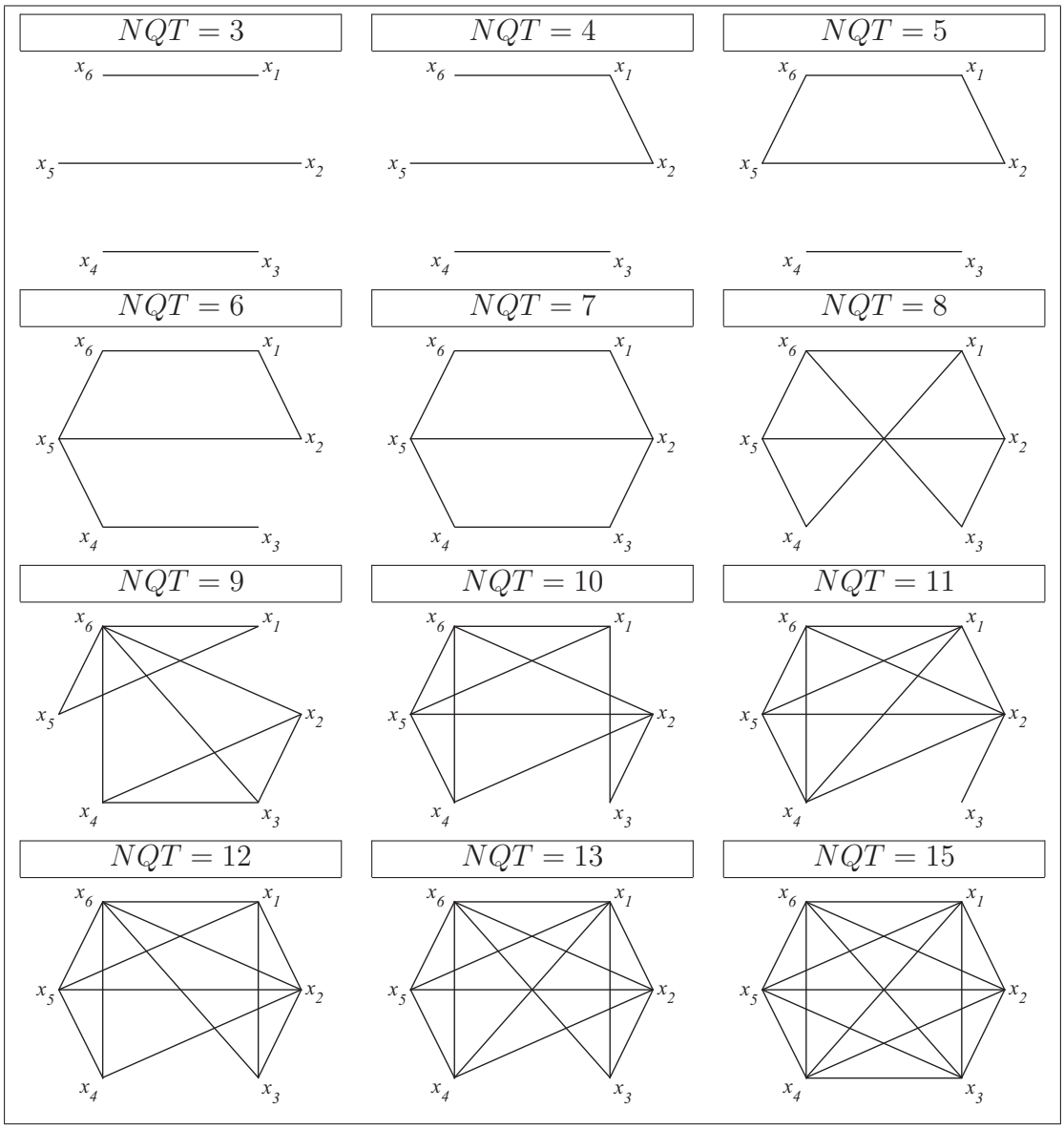


Figure 2.7: Graph representation of quadratic functions $n = 6$, $NL = 28$.

NQT	NLP_1	NLP_2	NLP_3	NLP_4	NLP_5
3	24.57	9.71	3.25	0.85	0.19
4	25.14	10.47	3.82	1.08	0.23
5	25.71	11.23	4.4	1.31	0.28
6	26.85	12	4.57	1.42	0.33
7	28	12	4.8	1.6	0.38
8	28	12	4.91	1.65	0.42
9	28	12	5.08	1.77	0.47
10	28	12	4.97	1.77	0.52
11	28	12	5.02	1.82	0.57
12	28	12	5.37	2	0.61
13	28	12	5.54	2	0.66
14	28	12	5.2	2	0.71
15	27.42	12	5.25	2	0.76
16	28	12	5.14	2	0.80
17	28	12	5.31	2	0.85
18	26.85	12	5.14	2	0.90
19	26.85	12	5.31	2	0.952
20	25.14	12	5.42	2	1
21	**28**	**12**	**6**	**2**	**1**

Table 2.4: The best nonlinearity profiles for 7-variables quadratic functions.

2.3 Linear structure of cryptographic rotation symmetric Boolean functions

Most modern encryption schemes achieve their security based on the properties of the corresponding Boolean functions. This is true for the overall plaintext-ciphertext mappings, as well as for the internal operations used, such as the substitution boxes in the round mapping of a product cipher. Recent advances in cryptanalysis techniques, for example differential [28] and linear [29] cryptanalysis show explicitly that weaknesses in the internal operations of a cipher can be exploited in attacks that succeed faster than exhaustive search. On the other hand, obtaining Boolean functions with strong properties is a non-trivial task due to the huge search space.

Linear structure is an important criterion to measure the weakness of Boolean functions in their cryptographic applications. This criterion has been investigated for its cryptanalytic

significance. In [32], Evertse stated that "a block cipher has a linear structure if there are subsets of plaintext bits P , key bits K and ciphertext bits C of this block cipher, such that for each plaintext and each key, a simultaneous change of all plaintext bits in P and all key bits in K has the same effect on the exclusive-or sum of the bits in C of the corresponding ciphertext." A good example on the threat of the existence of linear structures in block ciphers was given in [33], where the authors showed that the complementation of all plaintext bits and key bits in DES function results in the complementation of all ciphertext bits. Also, it has been shown in [22, 32–34] that block ciphers with linear structure are vulnerable to the attacks much faster than exhaustive search. Several studies were conducted on the existence of the linear structures in other classes of the Boolean functions (e.g., for vectorial functions (functions with multi outputs) [35] and for symmetric functions [36]).

2.3.1 Rotation symmetric Boolean functions

To solve the problem of the large search space, the space of the Boolean function was divided into several classes with smaller space. Rotation Symmetric Boolean Functions (RSBFs) is one of these rich cryptographic classes of Boolean functions which has been given more attention recently. RSBFs have been analyzed in [5], where the authors examined the nonlinearity of these functions up to 9 variables and found encouraging results. Then extended studies have been conducted in [6, 37] and significant properties of these functions up to 8 variables have been exhibited. The authors also discussed the enumeration of RSBFs for specific degrees. The usage of these functions was first carried out in [38], where they used it as components in the compression function of the hashing algorithm.

Definition 2.3.1. A Boolean function $f(\mathbf{x})$ is called RSBF if for each input $(x_1, \ldots, x_n) \in \mathbb{F}_2^n$, $(\rho_n^k(x_1, \ldots, x_n))=f(x_1, \ldots, x_n)$ for $1 \leq k \leq n$ where ρ_n^k acts as k-cyclic rotation on

an n-bit vector, i.e.

$$\rho_n^k(x_1, x_2, \ldots, x_n) = (x_{1+k}, x_{2+k}, ..., x_n, x_1, \ldots, x_k) \tag{2.4}$$

The space of RSBFs for n-variable is of a size approximately $2^{2^n/n}$, therefore the search in this space becomes comparatively easier than searching the whole space 2^{2^n}. It was shown in [6] that it is easy to find a RSBF with 7-variable, 2-resilient and nonlinearity 56, which was earlier considered as a function that is hard to search for. In addition, these functions also posses the best known autocorrelation spectra [39]. Besides, some of the Boolean functions that were open problems for quite some time have been found in the RSBF class or a concatenation of two RSBFs [5, 40]. Recently, the search in RSBF class has yielded Boolean function with high nonlinearity and maximum possible algebraic immunity [41] which can be used directly in stream ciphers. Thus it is important to study these functions and its possessed properties intensively.

The following example shows the partitions of the RSBF output to form smaller search space.

Example 2.3.1. For $n = 4$, the 16 input patterns of a RSBF are divided into 6 partitions which consequently divided the function output into the following subsets

$$f(0,0,0,0)$$
$$f(0,0,0,1) = f(0,0,1,0) = f(0,1,0,0) = f(1,0,0,0)$$
$$f(0,0,1,1) = f(0,1,1,0) = f(1,1,0,0) = f(1,0,0,1)$$
$$f(0,1,0,1) = f(1,0,1,0)$$
$$f(0,1,1,1) = f(1,0,1,1) = f(1,1,0,1) = f(1,1,1,0)$$
$$f(1,1,1,1)$$

Table 2.5 shows the number of RSBFs versus the total space for different values of n. By following the notations in [6], we denote by $G_n(x_1, \ldots, x_n)$ the orbit of $(x_1, \ldots, x_n)$

n	1	2	3	4	5	6	7	8	9	10
g_n	2	3	4	6	8	14	20	36	60	108
Total space	2	4	8	16	32	64	128	256	512	1024

Table 2.5: Number of RSBFs versus the total space for different values of n.

where $G_n(x_1, \ldots, x_n) = \{\rho_n^k(x_1, \ldots, x_n),\ 1 \leq k \leq n\}$, g_n is the number of the output partition and 2^{g_n} is the number of the RSBFs for n-variable function. Given $(x_1, \ldots, x_n)$, a function is RSBF if it takes the same value for all the inputs in $G_n(x_1, \ldots, x_n)$. From Example 2.3.1, $f(\rho_n^k(0,1,0,1)) = f(0,1,0,1) = f(1,0,1,0)$, $G_n(0,1,0,1) = G_n(\rho_n^k(0,1,0,1)) = \{(0,1,0,1),(1,0,1,0)\}$ and $g_4 = 6$.

Remark 2.1. The Walsh-transform of a rotation symmetric function is also a rotation symmetric.

2.3.2 Linear structures of rotation symmetric Boolean functions

Recall that a vector $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ is called a *linear structure* of $f(\mathbf{x})$ if$f(\mathbf{x} \oplus \boldsymbol{\alpha}) \oplus f(\mathbf{x}) = c$, where $c \in \mathbb{F}_2$. We denote by $\mathbf{0}$ and $\mathbf{1}$ the all-zero vector and all-one vector of $\mathbb{F}_2^n$ respectively. In this section, we first study the RSBF when its input bitwise XORed with all-one vector $\mathbf{1}$. Later we examine the existence of linear structures in RSBFs class.

Lemma 2.1. *If $f(\mathbf{x})$ is a RSBF then $f(\mathbf{x} \oplus \mathbf{1})$ is also RSBF.*

Proof. The proof follows by noting that $\rho_n^k(\mathbf{1}) = \mathbf{1}$, $1 \leq k \leq n$. Consequently, we have $f(\rho_n^k(\mathbf{x} \oplus \mathbf{1})) = f(\rho_n^k(\mathbf{x}) \oplus \rho_n^k(\mathbf{1}))$=$f(\mathbf{x} \oplus \mathbf{1})$.

□

The following lemma was proven in [23].

Lemma 2.2. *Let $f(\mathbf{x})$ be a Boolean function and $\mathbf{b} \in \mathbb{F}_2^n$. If $g(\mathbf{x}) = f(\mathbf{x} \oplus \mathbf{b})$, then*

$$deg(g(\mathbf{x})) = deg(f(\mathbf{x})) \tag{2.5}$$

Remark 2.2. $f(\mathbf{x})$ is balanced if and only if $F(\mathbf{0}) = 0$. Recall that if $g(\mathbf{x}) = f(\mathbf{x} \oplus \boldsymbol{\alpha})$, then the Walsh transform of $g(\mathbf{x})$ is given by $G(\mathbf{w}) = (-1)^{<\mathbf{w}\cdot\boldsymbol{\alpha}>} F(\mathbf{w})$. It is also easy to show that if $f(\mathbf{x})$ is balanced, then $G(\mathbf{0}) = 0$, i.e. $g(\mathbf{x})$ is also balanced.

Theorem 2.3. *If $f(\mathbf{x})$ is a RSBF and has a linear structure $\boldsymbol{\alpha}$, then all the elements in $G_n(\boldsymbol{\alpha})$ are also linear structures of $f(\mathbf{x})$.*

Proof. Since $f(\mathbf{x}) = f(\rho_n^k(\mathbf{x}))$, it follows that $f(\mathbf{x} \oplus \boldsymbol{\alpha}) = f(\rho_n^k(\mathbf{x} \oplus \boldsymbol{\alpha}))= f(\rho_n^k(\mathbf{x}) \oplus \rho_n^k(\boldsymbol{\alpha}))$. Thus, all the sets generated by $\rho_n^k(\boldsymbol{\alpha})$ are linear structures of $f(\mathbf{x})$. □

Remark 2.3. The derivative of $f(\mathbf{x})$(on a non-constant function) reduces its algebraic degree for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n \backslash \{\mathbf{0}\}$,

$$deg(d_{\boldsymbol{\alpha}} f(\mathbf{x})) < deg(f(\mathbf{x})). \tag{2.6}$$

Remark 2.4. If $deg(f(\mathbf{x})) = n$, then for all $\boldsymbol{\alpha} \in \mathbb{F}_2^n \backslash \{\mathbf{0}\}$

$$deg(d_{\boldsymbol{\alpha}} f(\mathbf{x})) = n - 1. \tag{2.7}$$

Lemma 2.4. *The function $\underset{d=n}{f}(\mathbf{x})$ has no non trivial linear structure.*

Proof. $\underset{d=n}{f}(\mathbf{x})$ can be written in the form

$$\underset{d=n}{f}(\mathbf{x}) = \underset{d\leq n-1}{g}(\mathbf{x}) \oplus x_1 x_2 \cdots x_n$$

It follows from remarks 2.3 and 2.4 that $deg(d_{\boldsymbol{\alpha}} g(\mathbf{x})) \leq n-2$ and $deg(d_{\boldsymbol{\alpha}}(x_1 x_2 \cdots x_n))=n-1$. Thus $deg(d_{\boldsymbol{\alpha}} \underset{d=n}{f}(\mathbf{x})) \neq 0$.

□

Theorem 2.5. *Let $\underset{d=n-1}{f}(\boldsymbol{x})$ be a RSBF with a linear structure $\boldsymbol{\alpha}$, then $\boldsymbol{\alpha} \in \{\mathbf{0}, \mathbf{1}\}$.*

Proof. If $f(\mathbf{x})$ is RSBF of degree $= n-1$, then all the terms of weight $n-1$ are in the same orbit $G_n(0,1,\ldots,1,1)$. As a result of the rotation, all the monomials of degree $n-1$ will appear in its ANF_f, and every variable x_i for $1 \leq i \leq n$ will appear exactly in $n-1$ monomials of degree $n-1$. Then we can write $\underset{d=n-1}{f(\mathbf{x})}$ in the following form,

$$\underset{d=n-1}{f(\mathbf{x})} = \underset{d<n-1}{g(\mathbf{x})} \oplus x_2x_3\cdots x_n \oplus x_1x_3\cdots x_n \oplus x_1x_2x_4\cdots x_n \oplus \cdots \oplus x_1x_2\cdots x_{n-1} \quad (2.8)$$

The derivative of any monomial of degree $n-1$, where it is degenerated in variable x_i , is not constant for all $\boldsymbol{\alpha}$ except for the vector with the i^{th} component equal 1 and all the others components are 0. By taking the derivative of the term $x_2x_3\cdots x_n \oplus x_1x_3\cdots x_n \oplus x_1x_2x_4\cdots x_n \oplus \cdots \oplus x_1x_2\cdots x_{n-1}$ for any $\boldsymbol{\alpha} \notin \{\mathbf{0},\mathbf{1}\}$, only one monomial will result in constant whereas the others will result in a *non-constant* distinctive terms of degree $n-2$. Also, the derivative of $\underset{d<n-1}{g(\mathbf{x})}$ will result in a function of degree less than $n-2$. Thus, the derivative of $\underset{d=n-1}{f(\mathbf{x})}$ will always result in a function of degree $n-2$.

□

Let $ANF_{G_n(x_1,\ldots,x_n)}$ denotes all the terms in the ANF generated by $G_n(x_1,\ldots,x_n)$.

Example 2.3.2. Let $f(\mathbf{x})$ be a RSBF with 3 variables where the orbit $G_n(0,1,1) = 1$ and the other orbits are zeros. Then $f(0,1,1) = f(1,1,0) = f(1,0,1) = 1$. Thus

$$ANF_f = ANF_{G_n(0,1,1)} = x_1x_2 \oplus x_2x_3 \oplus x_1x_3.$$

It is apparent that all x_i for $1 \leq i \leq 3$ appear exactly in $(n-1) = 2$ terms of degree $(n-1) = 2$. Thus, the derivative for this ANF_f will not be constant for any $\boldsymbol{\alpha} \in F_2^n$.

The following conjectures were confirmed by computer simulations for $n \leq 10$.

Conjecture 2.1. If $f(\mathbf{x})$ is a RSBF with even number of variables, balanced and has algebraic degree $d = n-1$, then $f(\mathbf{x})$ has no non trivial linear structure.

Conjecture 2.2. If $f(\mathbf{x})$ is RSBF with odd number of variables and has algebraic degree $d = n - 2$, then $f(\mathbf{x})$ has no non trivial linear structure.

2.4 Generalization of cryptographic Boolean functions properties to $GF(p)$

The existence of a tradeoff between the cryptographic properties in $GF(2)$ functions has an immense consequences on the security of the cryptosystem using these functions. For instance, the algebraic degree and the correlation immunity order in Boolean functions are two important security measures. It is well known that a cryptographic function that has a high resistance to correlation attacks may have a low linear complexity to counter the linear synthesis by the Berlekamp-Massey algorithm [42].

In the special case where $p = 2$, the Siegenthaler inequality [43] states that if a function $f(\mathbf{x})$ with n variables is a correlation-immune of order m then its algebraic degree $d \leq n - m$. Moreover, if $f(\mathbf{x})$ is an m-resilient, $m \leq n - 2$, then $d \leq n - m - 1$. It is clear from the Siegenthaler inequality that we cannot construct a function over $GF(2)$ with the maximum order of correlation immunity $(n - 1)$ and algebraic degree higher than 1. On the other hand, when the function is defined over $GF(p)$, it is possible to construct an $(n-1)$-correlation immune function with algebraic degree greater than 1. For example, let $f(\mathbf{x}) : \mathbb{F}_5^2 \rightarrow \mathbb{F}_5$ such that $f(x_1, x_2) = x_1 + x_2^3$. Then, $f(\mathbf{x})$ is a resilient function of degree 1 and its algebraic degree equals 3 [44].

This example illustrate the fact that functions over $GF(p)$ can possess high correlation immunity and high algebraic degree. Thus motivated by the better bounds these functions can achieve, various cryptographic properties have already been extended from $GF(2)$ to

other finite fields. For example, [44] presented a series of constructions of correlation-immune function over finite fields. Later, [45] investigated the existence, construction, and enumeration of resilient functions. [46] extended the concept of the Strict Avalanche Criterion (SAC) to $GF(p)$ functions. Due to its importance in cryptography and coding theory, bent function and its properties were generalized in [47]. The concept of hyper-bent function was extended to functions over $GF(p)$ in [48]. A new characterization of semi-bent and bent quadratic functions on finite fields was given in [49]. The author in [50] generalized the counting results of rotation symmetric Boolean functions to the rotation symmetric polynomials over finite fields $GF(p)$. [51] gave a lower bound on the number of n-variable balanced symmetric polynomials over finite fields $GF(p)$. In this section, we generalize some of the previous results on cryptographic binary functions to functions defined over $GF(p)$, where p is an odd prime.

The rest of this section is organized as follows. Section 2.4.1 briefly recalls the necessary definitions and algebraic preliminaries required in presenting our result. In section 2.4.2, we first generalize the Siegenthaler's construction to functions defined over finite field and then derive their cryptographic properties. In section 2.4.3, we characterize the linear structures of functions over $GF(p)$ in terms of their Walsh transform values. In section 2.4.4, we investigate the relation between the autocorrelation coefficients of functions over $GF(p)$ and their Walsh spectrum. In section 2.4.5, we derive the upper bound for the dimension of the linear space of the functions defined over $GF(p)$. Finally, in section 2.4.6 we present a method to construct bent functions from semi-bent functions that has linear structures. Throughout the rest of this chapter, let $f(\cdot)$ denotes an n variable function over $GF(p)$.

2.4.1 Preliminaries

In this section, we present some definitions and algebraic preliminaries required to prove our result.

If $f : \mathbb{F}_p^n \to \mathbb{F}_p$, then f can be uniquely expressed in the following form

$$f(x_1, x_2, \ldots, x_n) = \sum_{i_1=0}^{p-1} \sum_{i_2=0}^{p-1} \cdots \sum_{i_n=0}^{p-1} a_{i_1 i_2 \cdots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}, \tag{2.9}$$

where $a_{i_1 i_2 \cdots i_n} \in \mathbb{F}_p$. This representation of f is called the *algebraic normal form* of f. The largest $i_1 + i_2 + \cdots + i_n$ with $a_{i_1 i_2 \cdots i_n} \neq 0$ is called the *algebraic degree* of f. The function f is called *balanced* if its output is uniformly distributed.

Definition 2.4.1. Let p be a prime and $u = e^{i(2\pi/p)}$ be the q-th root unity in C, where $i = \sqrt{-1}$. The Walsh transform of a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ is defined as follows

$$F(\mathbf{w}) = \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}) - <\mathbf{w} \cdot \mathbf{x}>} \tag{2.10}$$

The autocorrelation function is defined as

$$AC(\boldsymbol{\alpha}) = \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}+\boldsymbol{\alpha}) - f(\mathbf{x})}, \tag{2.11}$$

where $\mathbf{w}, \boldsymbol{\alpha} \in \mathbb{F}_p^n$ and $< \mathbf{w} \cdot \mathbf{x} >$ denotes the dot product between $\mathbf{w}$ and $\mathbf{x}$, i.e., $< \mathbf{w} \cdot \mathbf{x} >= \sum_{i=1}^{n} w_i x_i \bmod p$. We will denote by $|X|$ the magnitude of the complex number X. Most of the properties of the cryptographic functions can be measured using the Walsh transform or the autocorrelation function.

Definition 2.4.2. A function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ is bent if and only if $|F(\mathbf{w})| = p^{n/2}$ for all $\mathbf{w} \in \mathbb{F}_p^n$ [47].

Definition 2.4.3. A function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ is semi-bent if and only if the absolute values of its Walsh transform are $|p^{(n+1)/2}|$ and 0 that occur with frequency p^{n-1} and $p^n - p^{n-1}$, respectively.

Definition 2.4.4. The derivative of a function $f(\mathbf{x})$ with respect to a vector $\mathbf{e} \in \mathbb{F}_p^n$ is defined as $d_{\mathbf{e}}f(\mathbf{x}) = f(\mathbf{x}+\mathbf{e}) - f(\mathbf{x})$. The vector $\mathbf{e}$ is called a *linear structure* of $f(\mathbf{x})$ if $d_{\mathbf{e}}f(\mathbf{x}) = c$ (constant) for any $\mathbf{x} \in \mathbb{F}_p^n$. The set of all linear structures of $f(\mathbf{x})$ form a subspace called *linear subspace* V_n.

2.4.2 Generalization of Siegenthaler's construction

A simple and useful method to construct Boolean functions is through direct constructions. Direct constructions can produce functions that are optimal with respect to the designed property. Lots of research efforts have been put into these construction techniques in $GF(2)$. Thus, it is significant to extend these constructions from $GF(2)$ to $GF(p)$. Siegenthaler [43] proposed a method to construct a Boolean function f of order n by combining two functions f_1, f_2 of order $n-1$, such that $f : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2 : (\bar{\mathbf{x}}, x_n) \mapsto (x_n \oplus 1)f_1(\bar{\mathbf{x}}) \oplus x_n f_2(\bar{\mathbf{x}})$, where $\bar{\mathbf{x}} = (x_1, \cdots, x_n - 1)$.

In this section we generalize the Siegenthaler's construction method to functions over $GF(p)$. We also derive some cryptographic properties of the constructed functions.

Let $f_1, f_2, \cdots, f_p : \mathbb{F}_p^{n-1} \to \mathbb{F}_p$. Consider a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ where $f = [f_1 \parallel f_2 \parallel \cdots \parallel f_p]$. In other words, f denotes the function whose truth table is the concatenation of the truth tables of $f_1, f_2, \cdots, f_p$ in the given order.

Algebraic Normal Form (ANF):

Let $\bar{\mathbf{x}} = (x_1, x_2, \ldots, x_{n-1})$ and $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1}, x_n)$, then,

$$\begin{aligned} f(\mathbf{x}|x_n = 0) &= f_1(\bar{\mathbf{x}}) \\ f(\mathbf{x}|x_n = 1) &= f_2(\bar{\mathbf{x}}) \\ &\vdots \\ f(\mathbf{x}|x_n = p-1) &= f_p(\bar{\mathbf{x}}) \end{aligned}$$

Then we can write the ANF of $f(\mathbf{x})$ as follows

$$f(\mathbf{x}) = (p-1)f_1(\bar{\mathbf{x}})\prod_{j=1}^{p-1}(x_n - j) + (p-1)f_2(\bar{\mathbf{x}})\prod_{\substack{j=0 \\ j\neq 1}}^{p-1}(x_n - j) + \cdots + (p-1)f_p(\bar{\mathbf{x}})\prod_{j=0}^{p-2}(x_n - j)$$

$$= \sum_{i=1}^{p}(p-1)f_i(\bar{\mathbf{x}}) \prod_{\substack{j=0 \\ j\neq (i-1)}}^{p-1}(x_n - j) \tag{2.12}$$

Walsh Transform: Let $\bar{\mathbf{w}} = (w_1, w_2, \ldots, w_{n-1})$ and $\mathbf{w} = (w_1, w_2, \ldots, w_{n-1}, w_n)$

$$\begin{aligned} f_1(\bar{\mathbf{x}}) &= f(\mathbf{x}|x_n = 0) \\ f_2(\bar{\mathbf{x}}) &= f(\mathbf{x}|x_n = 1) \\ &\vdots \\ f_p(\bar{\mathbf{x}}) &= f(\mathbf{x}|x_n = p-1) \end{aligned}$$

The Walsh transform of the concatenated function is given by

$$\begin{aligned} F(\mathbf{w}) &= \sum_{\mathbf{x}\in\mathbb{F}_p^n} u^{f(\mathbf{x}) - <\mathbf{w}\cdot\mathbf{x}>} \\ &= \sum_{\mathbf{x}|x_n=0} u^{f_1(\mathbf{x}) - <\mathbf{w}\cdot\mathbf{x}>} + \sum_{\mathbf{x}|x_n=1} u^{f_2(\mathbf{x}) - <\mathbf{w}\cdot\mathbf{x}>} + \cdots + \sum_{\mathbf{x}|x_n=p-1} u^{f_p(\mathbf{x}) - <\mathbf{w}\cdot\mathbf{x}>}. \end{aligned}$$

By noting that, $< \mathbf{w} \cdot \mathbf{x} >=< \bar{\mathbf{w}} \cdot \bar{\mathbf{x}} > + w_n x_n$, then

$$\begin{aligned} F(\mathbf{w}) &= \sum_{\bar{\mathbf{x}}} u^{f_1(\bar{\mathbf{x}}) - <\bar{\mathbf{w}} \cdot \bar{\mathbf{x}}>} + u^{-w_n} \sum_{\bar{\mathbf{x}}} u^{f_2(\bar{\mathbf{x}}) - <\bar{\mathbf{w}} \cdot \bar{\mathbf{x}}>} + \cdots + u^{-(p-1)w_n} \sum_{\bar{\mathbf{x}}} u^{f_p(\bar{\mathbf{x}}) - <\bar{\mathbf{w}} \cdot \bar{\mathbf{x}}>} \\ &= F_1(\bar{\mathbf{w}}) + u^{-w_n} F_2(\bar{\mathbf{w}}) + \cdots + u^{-(p-1)w_n} F_p(\bar{\mathbf{w}}) \\ &= \sum_{i=1}^{p} u^{(1-i)w_n} F_i(\bar{\mathbf{w}}). \end{aligned}$$

2.4.3 Characterization of linear structures of functions over $GF(p)$

Direct use of Boolean functions possessing linear structure should be avoided in cryptographic applications. It has been shown in [22, 32–34] that block ciphers with linear structure are vulnerable to attacks much faster than the exhaustive search. Several studies were conducted on the existence of the linear structures in several classes of Boolean functions, as in [35] for vectorial functions and for symmetric functions [36]. In this section, we study this criterion for functions defined over $GF(p)$. In particular, we characterize linear structures of functions over $GF(p)$ in terms of their Walsh transform values.

Theorem 2.6. *(Generalization of Theorem 1 in [35])* $f(\mathbf{x})$ *has a linear structure* $\mathbf{e} \in \mathbb{F}_p^n$ *with a corresponding constant* c *if and only if* $F(\mathbf{w}) = 0$ *for all* $\mathbf{w}$ *such that* $< \mathbf{w} \cdot \mathbf{e} >\neq c$.

Proof. Since $\mathbf{e}$ is a linear structure of $f(\mathbf{x})$, then $f(\mathbf{x} + \mathbf{e}) - f(\mathbf{x}) = c$, $c \in \mathbb{F}_p$. Let $g(\mathbf{x}) = f(\mathbf{x} + \mathbf{e}) - c$, then $G(\mathbf{w}) = F(\mathbf{w})$

$$\begin{aligned} G(\mathbf{w}) &= \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}+\mathbf{e}) - c - <\mathbf{x} \cdot \mathbf{w}>} \\ &= \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}) - c - <(\mathbf{x}-\mathbf{e}) \cdot \mathbf{w}>} \\ &= \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}) - c - <\mathbf{x} \cdot \mathbf{w}> + <\mathbf{w} \cdot \mathbf{e}>} \\ &= u^{<\mathbf{w} \cdot \mathbf{e}> - c} F(\mathbf{w}), \end{aligned}$$

Thus, **e** is a linear structure of $f(\mathbf{x})$ if and only if $f(\mathbf{x}) = g(\mathbf{x})$. which implies that $< \mathbf{w} \cdot \mathbf{e} > -c = 0$. □

We use Theorem 2.6 to characterize the linear structures of semi-bent functions defined over $GF(p)$.

Corollary 2.1. For a semi-bent function $f(\mathbf{x})$, $\mathbf{e}$ is a linear structure with a corresponding constant c if and only if $F(\mathbf{w}) = 0$ for all $\mathbf{w}$ such that $< \mathbf{w} \cdot \mathbf{e} > \neq c$ and $|F(\mathbf{w})| = p^{(n+1)/2}$ for all $\mathbf{w}$ such that $< \mathbf{w} \cdot \mathbf{e} > = c$.

Proof. The absolute value of the Walsh transform of the semi-bent function have only two values 0 and $p^{(n+1)/2}$. Since the number of $\mathbf{w}$ that satisfy the equation $< \mathbf{w} \cdot \mathbf{e} > \neq c$ is $p^{n-1}(p-1)$, which it is exactly the same number of zeros in the Walsh transform $F(\mathbf{w}) = 0$. Hence, there is a one-to-one mapping between the Walsh transform and the relation $< \mathbf{w} \cdot \mathbf{e} > \neq c$, i.e., $F(\mathbf{w}) = 0$ if and only if $< \mathbf{w} \cdot \mathbf{e} > \neq c$ and also $|F(\mathbf{w})| = p^{(n+1)/2}$ if and only if $< \mathbf{w} \cdot \mathbf{e} > = c$. □

2.4.4 Relation between the autocorrelation function and the Walsh transform

The autocorrelation is another useful criterion in analyzing Boolean functions. It measures the probability distribution of the output difference of the function for a fixed input difference. The autocorrelation coefficient $AC(\boldsymbol{\alpha})$ measures the statistical bias of the output distribution of $d_{\boldsymbol{\alpha}} f(\mathbf{x})$ relative to the uniform distribution. In this section we show how the autocorrelation coefficients of functions over $GF(p)$ are related to their Walsh spectrum.

Lemma 2.7. *Let $f(\mathbf{x})$ be a function defined over $GF(p)$. Then*

$$AC(\boldsymbol{\alpha}) = \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} |F(\mathbf{w})|^2 u^{<\mathbf{w} \cdot \boldsymbol{\alpha}>} \tag{2.13}$$

Proof. Using the inverse of the Walsh transform in equation 2.10, we get

$$u^{f(\mathbf{x})} = \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot\mathbf{x}>}.$$

Thus

$$\begin{aligned} u^{f(\mathbf{x}+\boldsymbol{\alpha})} &= \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\alpha})>} \\ &= \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot\mathbf{x}>} u^{<\mathbf{w}\cdot\boldsymbol{\alpha}>}. \end{aligned}$$

From the definition of the autocorrelation function in equation 2.11, we get

$$\begin{aligned} AC(\boldsymbol{\alpha}) &= \frac{1}{p^n} \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{-f(\mathbf{x})} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot\mathbf{x}>} u^{<\mathbf{w}\cdot\boldsymbol{\alpha}>} \\ &= \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot\boldsymbol{\alpha}>} \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{-f(\mathbf{x})} u^{<\mathbf{w}\cdot\mathbf{x}>} \\ &= \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot\boldsymbol{\alpha}>} \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{-f(\mathbf{x})+<\mathbf{w}\cdot\mathbf{x}>} \\ &= \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} F(\mathbf{w}) u^{<\mathbf{w}\cdot\boldsymbol{\alpha}>} F^*(\mathbf{w}), \end{aligned}$$

where $F^*(\mathbf{w})$ is the complex conjugate of $F(\mathbf{w})$. Then we have

$$AC(\boldsymbol{\alpha}) = \frac{1}{p^n} \sum_{\mathbf{w} \in \mathbb{F}_p^n} |F(\mathbf{w})|^2 u^{<\mathbf{w}\cdot\boldsymbol{\alpha}>}.$$

□

The following corollary follows directly from the definition of the inverse Walsh transform and Lemma 2.7.

Corollary 2.2. Let $f(\mathbf{x})$ be a function defined over $GF(p)$. Then

$$|F(\mathbf{w})|^2 = \sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} AC(\boldsymbol{\alpha}) u^{-<\mathbf{w}\cdot\boldsymbol{\alpha}>} \tag{2.14}$$

Lemma 2.8. *Let $f(\mathbf{x})$ be a function defined over $GF(p)$. Then*

$$\sum_{\mathbf{w}\in\mathbb{F}_p^n} |F(\mathbf{w})|^4 = p^n \sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}) \tag{2.15}$$

Proof. Squaring both sides of the equation in Corollary 2.2 we get

$$|F(\mathbf{w})|^4 = \sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} AC(\boldsymbol{\alpha}) u^{-<\boldsymbol{\alpha}\cdot\mathbf{w}>} \sum_{\boldsymbol{\beta}\in\mathbb{F}_p^n} AC(\boldsymbol{\beta}) u^{-<\boldsymbol{\beta}\cdot\mathbf{w}>}.$$

By taking the summation for both sides for all $w \in \mathbb{F}_p^n$ we get

$$\begin{aligned}\sum_{\mathbf{w}\in\mathbb{F}_p^n} |F(\mathbf{w})|^4 &= \sum_{\mathbf{w}\in\mathbb{F}_p^n} \sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} \sum_{\boldsymbol{\beta}\in\mathbb{F}_p^n} AC(\boldsymbol{\alpha}) AC(\boldsymbol{\beta}) u^{<(-\boldsymbol{\alpha}-\boldsymbol{\beta})\cdot\mathbf{w}>} \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} \sum_{\boldsymbol{\beta}\in\mathbb{F}_p^n} AC(\boldsymbol{\alpha}) AC(\boldsymbol{\beta}) \sum_{\mathbf{w}\in\mathbb{F}_p^n} u^{<(-\boldsymbol{\alpha}-\boldsymbol{\beta})\cdot\mathbf{w}>}.\end{aligned}$$

By noting that

$$\sum_{\mathbf{w}\in\mathbb{F}_p^n} u^{<(-\boldsymbol{\alpha}-\boldsymbol{\beta})\cdot\mathbf{w}>} = \begin{cases} 0 & \boldsymbol{\alpha} \neq -\boldsymbol{\beta} \\ p^n & \boldsymbol{\alpha} = -\boldsymbol{\beta} \end{cases},$$

then we have

$$\sum_{\mathbf{w}\in\mathbb{F}_p^n} |F(\mathbf{w})|^4 = p^n \sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}).$$

□

We now derive the relation between the Walsh spectrum of the semi-bent functions and

their autocorrelation coefficients.

Theorem 2.9. *Let $f(\mathbf{x})$ be a semi-bent function defined over $GF(p)$. Then*

$$p^n F_{max}^2(\mathbf{w}) = \sum_{\boldsymbol{\alpha} \in \mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}) \tag{2.16}$$

Proof. Since $f(\mathbf{x})$ is a semi-bent function, the Walsh transform contains the values $F_{max}(\mathbf{w})$ $= p^{(n+1)/2}$ and occurs p^{n-1} times while 0 occurs $(p^n - p^{n-1})$ times. We refer throughout the rest of the section to the value $p^{(n+1)/2}$ as $F_{max}(\mathbf{w})$. Thus

$$\sum_{\mathbf{w} \in \mathbb{F}_p^n} |F(\mathbf{w})|^4 = p^{n-1} F_{max}^4(\mathbf{w}) = p^{3n+1}.$$

Substituting in Lemma 2.8, we get

$$\begin{aligned} p^n \sum_{\boldsymbol{\alpha} \in \mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}) &= p^{3n+1} \\ \sum_{\boldsymbol{\alpha} \in \mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}) &= p^{2n+1} \\ &= p^n F_{max}^2(\mathbf{w}). \end{aligned}$$

□

2.4.5 Walsh spectrum of $GF(p)$ functions with linear structure

In this section we derive the upper bound of the dimension of the linear space of the functions defined over $GF(p)$.

Theorem 2.10. *(Generalization of theorem 3 in [52]) Let $f(\mathbf{x})$ be a function defined over $GF(p)$ with n variables. Then, the dimension k of the linear space V_n is such that $k \leq 1$.*

Proof.

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}) = \sum_{\boldsymbol{\alpha}\in V} AC^2(\boldsymbol{\alpha}) + \sum_{\boldsymbol{\alpha}\notin V} AC^2(\boldsymbol{\alpha}).$$

If $f(\mathbf{x})$ has a linear space of dimension k, then

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_p^n} AC^2(\boldsymbol{\alpha}) = p^k p^{2n} + \sum_{\boldsymbol{\alpha}\notin V} AC^2(\boldsymbol{\alpha}) \geq p^{2n+k}.$$

Substituting in Theorem 2.9 we get

$$p^n F_{max}^2(\mathbf{w}) \geq p^{2n+k}.$$

Thus

$$F_{max}(\mathbf{w}) \geq p^{\frac{n+k}{2}}.$$

For a semi-bent function $F_{max}(\mathbf{w}) = p^{\frac{n+1}{2}}$, then

$$p^{\frac{n+1}{2}} \geq p^{\frac{n+k}{2}},$$

which implies that $k \leq 1$. □

2.4.6 Construction of bent functions from semi-bent functions with linear structure

Bent functions achieve the best possible nonlinearity. Accordingly, they provide good confusion properties, and they are perfect in resisting differential cryptanalysis [28] and by definition linear cryptanalysis [29]. Their major flaw is that they are not balanced. Another useful class of functions which achieve high nonlinearity is semi-bent functions. These functions also possess good cryptographic characteristics, and some of them are balanced.

Bent and semi-bent functions over $GF(p)$, $p > 2$, can exist in even and odd dimensions. It is possible to construct bent functions with $(n+1)$ variables from semi-bent function with n variables, and similarly, construct semi-bent functions with n variables from bent functions with $(n+1)$ variables. Here, we focus on constructing bent functions with $n+1$ variables from semi-bent functions with n variables.

The following lemmas are needed to simplify the proof of Theorem 2.14.

Lemma 2.11. *Let $g(\mathbf{x}) = f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e} >$. If $\mathbf{e}$ is a linear structure for $f(\mathbf{x})$ with a corresponding constant c, then $g(\mathbf{x})$ has $\mathbf{e}$ as a linear structure with the corresponding constant $c - < \mathbf{e} \cdot \mathbf{e} >$.*

Proof. If $f(\mathbf{x}+\mathbf{e}) - f(\mathbf{x}) = c$ and $g(\mathbf{x}) = f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e} >$ then

$$\begin{aligned} g(\mathbf{x}+\mathbf{e}) - g(\mathbf{x}) &= f(\mathbf{x}+\mathbf{e}) - < (\mathbf{x}+\mathbf{e}) \cdot \mathbf{e} > - f(\mathbf{x}) + < \mathbf{x} \cdot \mathbf{e} > \\ &= f(\mathbf{x}+\mathbf{e}) - < \mathbf{x} \cdot \mathbf{e} > - < \mathbf{e} \cdot \mathbf{e} > - f(\mathbf{x}) + < \mathbf{x} \cdot \mathbf{e} > \\ &= f(\mathbf{x}+\mathbf{e}) - f(\mathbf{x}) - < \mathbf{e} \cdot \mathbf{e} > \\ &= c - < \mathbf{e} \cdot \mathbf{e} > \end{aligned}$$

□

Lemma 2.12. *If $g(\mathbf{x}) = f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e} >$ then $G(\mathbf{w}) = F(\mathbf{w}+\mathbf{e})$.*

Proof.

$$\begin{aligned} G(\mathbf{w}) &= \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}) - <\mathbf{x}\cdot\mathbf{e}> - <\mathbf{x}\cdot\mathbf{w}>} \\ &= \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}) - [<\mathbf{x}\cdot\mathbf{e}> + <\mathbf{x}\cdot\mathbf{w}>]} \\ &= \sum_{\mathbf{x} \in \mathbb{F}_p^n} u^{f(\mathbf{x}) - <\mathbf{x}\cdot(\mathbf{w}+\mathbf{e})>} \\ &= F(\mathbf{w}+\mathbf{e}) \end{aligned}$$

□

Lemma 2.13. *If $f(\mathbf{x})$ has linear structures $\mathbf{a}$ and $\mathbf{b}$ with corresponding constants c_1 and c_2, respectively. Then $\mathbf{e} = (e_1, e_2, \cdots, e_n) = \mathbf{a} \boxminus \mathbf{b}$ is a linear structure for $f(\mathbf{x})$ with a corresponding constants $c_1 - c_2$, where $e_i = a_i - b_i \mod p$, $1 \leq i \leq n$.*

Proof. Let $f(\mathbf{x}+\mathbf{e}_1) - f(\mathbf{x}) = c_1$ and $f(\mathbf{x}+\mathbf{e}_2) - f(\mathbf{x}) = c_2$. Then $f(\mathbf{x}+\mathbf{e}_1) - f(\mathbf{x}+\mathbf{e}_1) = c_1 - c_2$ and $f(\mathbf{x} + (\mathbf{e}_1 - \mathbf{e}_2)) - f(\mathbf{x}) = c_1 - c_2$, which implies $(\mathbf{e}_1 - \mathbf{e}_2)$ is a linear structure with a corresponding constant $c_1 - c_2$. □

From the above lemma, it follows that if $\mathbf{e}$ is a linear structure for $f(\mathbf{x})$, then $a\,\mathbf{e}$, $a \in \mathbb{F}_p$ is also a linear structure for $f(\mathbf{x})$, where $a\,\mathbf{e}$ denotes the vector whose coordinates are obtained by multiplying the individual coordinates of $\mathbf{e}$ by $a \mod p$.

Theorem 2.14. *Let $f(\mathbf{x})$ be a semi-bent function defined over $GF(p)$ with non trivial linear structures $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{p-1}$. Then*

$$[f(\mathbf{x})||f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e}_1 > ||f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e}_2 > || \ldots ||f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e}_{p-1} >]$$

is $n+1$ bent function if $< \mathbf{e}_i \cdot \mathbf{e}_i > \neq 0$, for all $i = 1, \cdots, p-1$.

Proof. Since $f(\mathbf{x})$ has linear structures $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{p-1}$ with corresponding constants $c_1, c_2, \cdots, c_{p-1}$, respectively, then from Lemmas 2.11 and 2.12, the function $f(\mathbf{x}) - < \mathbf{x} \cdot \mathbf{e}_i >$, $1 \leq i \leq p-1$, will have a linear structure $\mathbf{e}_i$ with a corresponding constant $c_i - < \mathbf{e}_i \cdot \mathbf{e}_i >$ and Walsh transform $F(\mathbf{w} + \mathbf{e}_i)$.

From Corollary 2.1, we have

$$\begin{aligned} F(\mathbf{w}) = 0 \quad &\Leftrightarrow \quad < \mathbf{w} \cdot \mathbf{e}_1 > \neq c_1, < \mathbf{w} \cdot \mathbf{e}_2 > \neq c_2, \cdots, < \mathbf{w} \cdot \mathbf{e}_{p-1} > \neq c_{p-1} \\ F(\mathbf{w}) = p^{(n+1)/2} \quad &\Leftrightarrow \quad < \mathbf{w} \cdot \mathbf{e}_1 > = c_1, < \mathbf{w} \cdot \mathbf{e}_2 > = c_2, \cdots, < \mathbf{w} \cdot \mathbf{e}_{p-1} > = c_{p-1} \end{aligned}$$

By noting that $< (\mathbf{w} + \mathbf{e}_i) \cdot \mathbf{e}_i >=< \mathbf{w} \cdot \mathbf{e}_i > + < \mathbf{e}_i \cdot \mathbf{e}_i >$, where $1 \leq i \leq p-1$, then

$$
\begin{array}{lcl}
F(\mathbf{w}+\mathbf{e}_1) = 0 & \Leftrightarrow & < \mathbf{w} \cdot \mathbf{e}_1 > + < \mathbf{e}_1 \cdot \mathbf{e}_1 > \neq c_1 - < \mathbf{e}_1 \cdot \mathbf{e}_1 > \\
|F(\mathbf{w}+\mathbf{e}_1)| = p^{(n+1)/2} & \Leftrightarrow & < \mathbf{w} \cdot \mathbf{e}_1 > + < \mathbf{e}_1 \cdot \mathbf{e}_1 > = c_1 - < \mathbf{e}_1 \cdot \mathbf{e}_1 > \\
F(\mathbf{w}+\mathbf{e}_2) = 0 & \Leftrightarrow & < \mathbf{w} \cdot \mathbf{e}_2 > + < \mathbf{e}_2 \cdot \mathbf{e}_2 > \neq c_2 - < \mathbf{e}_2 \cdot \mathbf{e}_2 > \\
|F(\mathbf{w}+\mathbf{e}_2)| = p^{(n+1)/2} & \Leftrightarrow & < \mathbf{w} \cdot \mathbf{e}_2 > + < \mathbf{e}_2 \cdot \mathbf{e}_2 > = c_2 - < \mathbf{e}_2 \cdot \mathbf{e}_2 > \\
 & \vdots & \\
F(\mathbf{w}+\mathbf{e}_{p-1}) = 0 & \Leftrightarrow & < \mathbf{w} \cdot \mathbf{e}_{p-1} > + < \mathbf{e}_{p-1} \cdot \mathbf{e}_{p-1} > \neq c_{p-1} - < \mathbf{e}_{p-1} \cdot \mathbf{e}_{p-1} > \\
|F(\mathbf{w}+\mathbf{e}_{p-1})| = p^{(n+1)/2} & \Leftrightarrow & < \mathbf{w} \cdot \mathbf{e}_{p-1} > + < \mathbf{e}_{p-1} \cdot \mathbf{e}_{p-1} > = c_{p-1} - < \mathbf{e}_{p-1} \cdot \mathbf{e}_{p-1} >
\end{array}
$$

Thus, if $< \mathbf{w} \cdot \mathbf{e}_1 >= c_1$ then $|F(\mathbf{w})| = p^{(n+1)/2}$, $F(\mathbf{w}+\mathbf{e}_1) = 0$, $F(\mathbf{w}+\mathbf{e}_2) = 0$, $\cdots$, $F(\mathbf{w}+\mathbf{e}_{p-1}) = 0$. Consequently, if one of the $|F(\mathbf{w})|$, $|F(\mathbf{w}+\mathbf{e}_1)|$, $|F(\mathbf{w}+\mathbf{e}_2)|$, $\cdots$, $|F(\mathbf{w}+\mathbf{e}_{p-1})|$ equals $p^{(n+1)/2}$ the others equal zero, which implies that $F(\mathbf{w})$ corresponds to the Walsh transform of an $n+1$ bent function. □

CHAPTER 3

Cryptanalysis of a Public Key Cryptosystem Based on Boolean Permutations

Several attempts were made to construct public key cryptosystems based on Boolean permutations. Wu and Varadharajan [53] proposed three such constructions (PKC1, PKC2 and PKC3) whose security is based on the difficulty of inverting a special class of trapdoor Boolean permutations that can be constructed efficiently. In this chapter, we analyze the security of the PCK2 family proposed by Wu and Varadharajan. In particular, we show that the suggested construction for the PCK2 family is insecure and we present an efficient cryptanalytic attack that allows the cryptanalyst to invert the class of Boolean permutations used in PCK2 without the knowledge of the secret key parameters.

3.1 Introduction

Traditional public key systems [1] can be categorized, based on the underlying hard problem, into discrete logarithm-based, such as Diffie-Hellman, DSA and their elliptic curve

variants, or integer factorization-based such as the RSA algorithm. Despite the significant progress that has been made in implementing these systems, the computational cost associated with them remain somewhat expensive, especially in resource constrained environments. For example, a typical public key algorithm might involve about 500,000 32-bit multiplications which can take up to 5 seconds on low-end chips [54].

To address this problem, several attempts were made to construct public key cryptosystems based on trapdoor Boolean permutations that can be implemented efficiently (e.g. [55]- [56]). Any cryptosystem, without expansion, can be seen as a Boolean permutation which can be seen as a collection of Boolean functions.

Generally, finding the inverse of a nonlinear Boolean permutation is an NP-complete problem, which is equivalent to solving a set of nonlinear Boolean equations. The main idea behind designing trapdoor Boolean permutations is to find a class of efficiently computable random-looking permutations with compact representation, which can be easily inverted given some trapdoor information and, yet, are hard to invert without this information. If such a trapdoor Boolean permutation is constructed, then it is easily to design a public key cryptosystem in which the trapdoor information corresponds to the receiver's private key.

In [53], Wu and Varadharajan proposed three families of Boolean permutation based public key cryptosystems (PKC1, PKC2 and PKC3). In this chapter, we focus on the security of PKC2. It should be noted that PKC1 was not fully specified in [53] which makes it hard to present a specific attack against it. Also, our attack does not apply to PKC3, since its security also relies on the difficulty of decoding random codes.

The rest of this chapter is organized as follows. In section 3.2, we review some preliminaries related to PKC2 and our attack against it. The relevant details of PKC2 are reviewed in section 3.3. Our proposed attack is presented in section 3.4 and then provide numerical example in section 3.5.

3.2 Construction of Boolean permutation

A mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called an (n, m)-Boolean function. An (n, m)-Boolean function can be expressed as a collection of m Boolean functions with n variables. A special class of this mapping called a *Boolean permutation* occurs when $m = n$ and yields a one-to-one mapping between the inputs and the outputs. A Boolean permutation of order n can be expressed as a collection of Boolean functions of n variables and it is written as

$$P(\mathbf{x}) = [p_1(\mathbf{x}), \cdots, p_n(\mathbf{x})]. \tag{3.1}$$

If all the functions $p_1(\mathbf{x}), \cdots, p_n(\mathbf{x})$ are of algebraic degree one, we call $P(\mathbf{x})$ a linear Boolean permutation.

Lemma 3.1. *Let* $l_i(\mathbf{x}) = a_{i_0} \bigoplus a_{i_1} x_1 \bigoplus \cdots \bigoplus a_{i_n} x_n$, $a_{i_j} \in \mathbb{F}_2$, $i = 1, \cdots, n$. *Let* $A = [a_{i_j}]$, $i, j = 1, \cdots, n$, *be the matrix of coefficients. Then* $L = [l_1, \cdots, l_n]$ *forms a linear Boolean permutation if and only if* A *is nonsingular.*

The following algorithm [53] can be used to construct a Boolean permutation of order $n + 1$ from a Boolean permutation of order n.

Algorithm 1

Let $P(\mathbf{x}) = [p_1(\mathbf{x}), \cdots, p_n(\mathbf{x})]$, $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$ be a Boolean permutation of order n. Set $q_i(\mathbf{x}, x_{n+1}) = p_i(\mathbf{x}) \oplus x_{n+1} \oplus g(\mathbf{x})$, $i = 1, \cdots, n$, and $q_{n+1}(\mathbf{x}, x_{n-1}) = x_{n+1} \oplus g(\mathbf{x})$, $x_n \in \mathbb{F}_2$. Then $Q = [q_1, \cdots, q_{n+1}]$ is a Boolean permutation of order $n + 1$.

Algorithm 1 is an iterative construction method which requires an initial Boolean permutation. One approach is to select an arbitrary Boolean function of small order as the initial one. Apparently, when n becomes larger this construction tends to be inefficient,

since the initial permutation of order n is still large and hard to present efficiently. Therefore, the authors in [53] suggested the use of a linear Boolean permutation to initialize Algorithm 1 since linear permutations are simple to construct for any order. The algorithm can be iterated several times until we reach to the required order. It should be noted that a different nonlinear function $g_i(\mathbf{x})$ is chosen for every algorithm run.

In general, finding the inverse of a nonlinear Boolean permutation is equivalent to solving a set of nonlinear Boolean equations which is an NP-complete problem. In [53], the authors presented an efficient algorithm to obtain the inverse of P constructed by Algorithm 1 given the knowledge of the initial permutation and the functions g_i's.

3.3 Description of PKC2 proposed by Wu and Varadharajan

Let **A** be an arbitrary binary matrix with dimensions $k \times n$ and $rank(\mathbf{A}) = k$. Choose two matrices **S** and **B** with dimension $n \times k$ and $(n-k) \times n$ respectively, such that

$$\mathbf{A}\,\mathbf{S} = \mathbf{I}_k \quad \text{and} \quad \mathbf{B}\,\mathbf{S} = \mathbf{0} \tag{3.2}$$

One way of obtaining such **S**, **A** and **B** [53] is to choose an arbitrary $n \times n$ nonsingular matrix **G**, then let **A** be composed of the first k rows of **G** and **B** be composed of the remaining $n-k$ rows and let **S** be composed of the first most left k columns of $\mathbf{G}^{-1}$.

By iterating Algorithm 1, construct a Boolean permutation $P = [p_1(\mathbf{x}), \cdots, p_k(\mathbf{x})]$ of order k, keeping its inverse P^{-1} as secret. Let $R = [r_1, \cdots, r_{n-k}]$ be a collection of arbitrary Boolean functions.

The tuple $(\mathbf{A}, \mathbf{B}, P)$ as constructed above combined with R is used to construct the public key as

$$Q = P\,\mathbf{A} \oplus R\,\mathbf{B} \tag{3.3}$$

The corresponding private key is given by $P^{-1}(\cdot)$. The ciphertext corresponding to a k-bit message $\mathbf{m}$ is given by

$$\mathbf{c} = Q(\mathbf{m}) \tag{3.4}$$

The decryption of the ciphertext $\mathbf{c}$ is given by

$$\mathbf{m} = P^{-1}(\mathbf{c}\,\mathbf{S}) \tag{3.5}$$

The public key Q is a set of n Boolean functions of k variables $[q_1(\mathbf{x}), \cdots, q_n(\mathbf{x})]$ while the private key is a collection of k Boolean functions of n variables. Figure. 3.1 depicts the encryption process. The reader is referred to [53] for further details about the PKC2 cryptosystem.

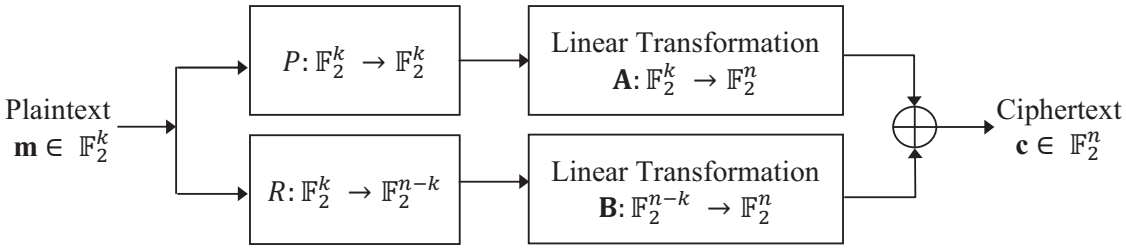


Figure 3.1: Block Diagram of PKC2 Encryption

3.4 The proposed attack

Recall that the ciphetext **c** corresponding to a message **m** is given by

$$\mathbf{c} = Q(\mathbf{m}) = P(\mathbf{m})\ \mathbf{A} \oplus R(\mathbf{m})\ \mathbf{B} \tag{3.6}$$

Multiplying both sides of the above equation by **S**, we obtain

$$\mathbf{c}\ S = P(\mathbf{m})\ \mathbf{A}\ \mathbf{S} \oplus R(\mathbf{m})\ \mathbf{B}\ \mathbf{S} = P(\mathbf{m})\ \mathbf{I} \oplus \mathbf{0} = P(\mathbf{m}). \tag{3.7}$$

The above equation implies that

$$\mathbf{c} \cdot \mathbf{s}_i^{tr} = p_i(\mathbf{m}), \quad i = 1, \cdots k, \tag{3.8}$$

where $\mathbf{s}_i^{tr}$ denotes the transpose of the i^{th} column of S. Let $\mathbf{m}' = (m_1, m_2, \cdots, m_{k-T})$. Assuming that P is constructed by iterating Algorithm 1 above for T times and starting from a linear permutation (see Lemma 3.1), then we can form the following system of $n - (T-1)$ equations

$$\mathbf{c} \cdot (\mathbf{s}_1 \oplus \mathbf{s}_i) = p_1(\mathbf{m}') \oplus p_i(\mathbf{m}') = (\mathbf{a}_1 \oplus \mathbf{a}_i) \cdot \mathbf{m}', \quad i = 2, \cdots n - T. \tag{3.9}$$

The above equation implies that certain linear combinations of the ciphertext bits can be expressed as a linear combination of the plaintext bits. It should be noted that the number of non zero terms in the ANF description of Q has to be kept small, otherwise the public key size will become impractical. Hence, given the ANF description of the public key, i.e., the ANF of q_i, $i = 1, \cdots n$, the Gaussian elimination (as shown by Algorithm *2* below) can be used to efficiently determine the subsets of the Q coordinates whose XOR yields a

linear function in the plaintext message bits.

Algorithm 2

---

(1) Let **W** denote the matrix whose i^{th} row corresponds to the nonlinear terms in the ANF representation of q_i. Let **O** denote the matrix obtained from **W** by removing all the zero columns from **W**.

(2) Let $\mathbf{L} = [\mathbf{O}|\mathbf{I}]$ where **I** denotes the identity matrix

(3) Run the Gaussian elimination algorithm to find the linearly dependent rows in **L**.

(4) Form a set of linear equations between **m** and **c** from the linearly dependent q_i's.

It should be noted that the above systems of equations can be formed off-line. Then we can solve for the plaintext corresponding to any given ciphertext by substituting the c_i's values into q_i's and solving for **m**. From (3.9), the rank, r, of the linear equations will be equal to $r = k - T$. This allows us to recover $k - T$ bits from the plaintext message. The remaining T bits can be recovered by assuming some values for them and checking the corresponding ciphertext using the known public key. It should be noted that in order to keep the size of the public key reasonably small, T has to be kept relatively small which consequently ensures the efficiency of our attack. A illustrative example that demonstrates the steps described in the above attack is given in the next section.

3.5 Numerical example

Consider PKC2 with $k = 5, n = 7$,

$$\mathbf{A} = \begin{bmatrix} 0100001 \\ 1011001 \\ 0101101 \\ 0100000 \\ 0100100 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0111011 \\ 0011100 \end{bmatrix}, \text{and } \mathbf{S} = \begin{bmatrix} 11001 \\ 00010 \\ 10100 \\ 10111 \\ 00011 \\ 10011 \\ 10010 \end{bmatrix}.$$

Let P be obtained by running Algorithm 1 for one time, starting from the linear permutation $l(x_1, \cdots, x_4) = [x_1, x_2, x_3, x_4]$ and using nonlinear function g as follows $g(x_1, \cdots, x_4) = x_1 \oplus x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_4 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_2x_3x_4$.

Let $R(\mathbf{x}) = [r_1(\mathbf{x}), r_2(\mathbf{x})]$ where

$$\begin{aligned} r_1(x_1, \cdots, x_4) = {} & x_1 \oplus x_4 \oplus x_1x_4 \oplus x_1x_2x_3x_4 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_3x_5 \oplus x_1x_3x_5 \oplus x_1x_2x_3x_5 \oplus \\ & x_2x_4x_5 \oplus x_1x_2x_4x_5. \\ r_2(x_1, \overset{x_4x_5\oplus}{\cdots}, x_4) = {} & 1 \oplus x_1 \oplus x_2 \oplus x_1x_2x_3 \oplus x_4 \oplus x_1x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4 \oplus x_5 \oplus x_3x_5 \oplus x_1x_3x_5 \oplus \\ & x_1x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_3x_4x_5. \end{aligned}$$

Then the ANF of the coordinates of the public key $Q = P\ \mathbf{A} \oplus R\ \mathbf{B}$ are given by

$$\begin{aligned} q_1(x_1, \cdots, x_5) = {} & x_1 \oplus x_2 \oplus x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_4 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_2x_3x_4 \oplus x_5. \\ q_2(x_1, \cdots, x_5) = {} & x_3 \oplus x_1x_4 \oplus x_1x_2x_3x_4 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_3x_5 \oplus x_1x_3x_5 \oplus x_1x_2x_3x_5 \oplus x_4x_5 \\ & \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5. \\ q_3(x_1, \cdots, x_5) = {} & 1 \oplus x_1 \oplus x_2x_3 \oplus x_1x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_1x_2x_3x_5 \oplus x_4x_5 \\ & \oplus x_1x_4x_5 \oplus x_1x_2x_4x_5 \oplus x_1x_3x_4x_5. \\ q_4(x_1, \cdots, x_5) = {} & 1 \oplus x_3 \oplus x_1x_2x_3 \oplus x_1x_3x_4 \oplus x_2x_3x_4 \oplus x_1x_2x_3x_4 \oplus x_5 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus \\ & x_1x_2x_3x_5 \oplus x_4x_5 \oplus x_1x_4x_5 \oplus x_1x_2x_4x_5 \oplus x_1x_3x_4x_5. \\ q_5(x_1, \cdots, x_5) = {} & 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_1x_2x_3 \oplus x_4 \oplus x_1x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4 \oplus x_5 \oplus x_3x_5 \oplus \\ & x_1x_3x_5 \oplus x_1x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_3x_4x_5. \end{aligned}$$

$$q_6(x_1,\cdots,x_5) = x_1\oplus x_4\oplus x_1x_4\oplus x_1x_2x_3x_4\oplus x_2x_5\oplus x_1x_2x_5\oplus x_3x_5\oplus x_1x_3x_5\oplus x_1x_2x_3x_5\oplus$$

$$x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5.$$

$$q_7(x_1,\cdots,x_5) = x_1\oplus x_2\oplus x_3\oplus x_2x_3\oplus x_1x_2x_3\oplus x_4\oplus x_1x_2x_4\oplus x_1x_3x_4\oplus x_5\oplus x_2x_5\oplus x_1x_2x_5$$

$$\oplus x_3x_5 \oplus x_1x_3x_5 \oplus x_1x_2x_3x_5 \oplus x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5.$$

Using Algorithm 2, our attack proceeds as follows:

(1) From the ANF of Q, we have

$$\mathbf{W} = \begin{array}{c} \begin{array}{ccccccccccccccccccccccccccccc} x_1x_2 & x_1x_3 & x_2x_3 & x_1x_2x_3 & x_1x_4 & x_2x_4 & x_1x_2x_4 & x_3x_4 & x_1x_3x_4 & x_2x_3x_4 & x_1x_2x_3x_4 & x_1x_5 & x_2x_5 & x_1x_2x_5 & x_3x_5 & x_1x_3x_5 & x_2x_3x_5 & x_1x_2x_3x_5 & x_4x_5 & x_1x_4x_5 & x_2x_4x_5 & x_1x_2x_4x_5 & x_3x_4x_5 & x_1x_3x_4x_5 & x_2x_3x_4x_5 & x_1x_2x_3x_4x_5 \end{array} \\ \begin{bmatrix} 0&0&1&1&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&1&0&1&1&1&1&0&1&1&0&1&1&0&0&0&0&0 \\ 0&0&1&0&1&0&1&0&0&1&0&0&1&1&0&0&0&1&1&1&0&1&0&1&0&0 \\ 0&0&0&1&0&0&0&0&1&1&1&0&1&1&0&0&0&1&1&1&0&1&0&1&0&0 \\ 0&0&0&1&1&0&0&0&1&1&0&0&0&0&1&1&0&0&0&1&1&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&0&0&1&0&1&1&1&1&0&1&1&0&1&1&0&0&0&0 \\ 0&0&1&1&0&0&1&0&1&0&0&0&1&1&1&1&0&1&1&0&1&1&0&0&0&0 \end{bmatrix} \end{array}$$

(2) After removing all the zero columns from **W**, we have

$$\mathbf{O} = \begin{bmatrix} 1&1&1&1&1&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&1&1&1&1&1&0&1&1&0 \\ 1&0&1&1&0&1&0&1&1&0&0&1&1&1&0&1&1 \\ 0&1&0&0&1&1&1&1&1&0&0&1&1&1&0&1&1 \\ 0&1&1&0&1&1&0&0&0&1&1&0&0&1&1&0&1 \\ 0&0&1&0&0&0&1&1&1&1&1&1&0&1&1&0 \\ 1&1&0&1&1&0&0&1&1&1&1&1&1&0&1&1&0 \end{bmatrix}$$

Note that in practice we do not need to construct **W**, i.e., only **O** needs to be constructed.

$$\mathbf{L} = \left[\begin{array}{cccccccccccccccc|ccccccc} 1&1&1&1&1&0&1&0&0&0&0&0&0&0&0&0 & 1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&1&1&1&1&1&1&0&1&0 & 0&1&0&0&0&0&0 \\ 1&0&1&1&0&1&0&1&1&0&0&1&1&1&0&1 & 0&0&1&0&0&0&0 \\ 0&1&0&0&1&1&1&1&1&0&0&1&1&1&0&1 & 0&0&0&1&0&0&0 \\ 0&1&1&0&1&1&0&0&0&1&1&0&0&1&1&0 & 0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&1&1&1&1&1&1&0&1&0 & 0&0&0&0&0&1&0 \\ 1&1&0&1&1&0&0&1&1&1&1&1&1&0&1&0 & 0&0&0&0&0&0&1 \end{array}\right]$$

(3) After running the Gaussian elimination we get

$$\begin{array}{c} \\ \\ \\ eq_1: \\ eq_2: \\ eq_3: \\ eq_4: \end{array} \left[\begin{array}{cccccccccccccccc|ccccccc} 1&1&1&1&1&0&1&0&0&0&0&0&0&0&0&0 & 1&0&0&0&0&0&0 \\ 0&1&0&0&1&1&1&1&1&0&0&1&1&1&0&1 & 1&0&1&0&0&0&0 \\ 0&0&1&0&0&0&1&1&1&1&1&1&1&0&1&0 & 0&1&0&0&0&0&0 \\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0} & \underline{\mathbf{1}}&0&\underline{\mathbf{1}}&\underline{\mathbf{1}}&0&0&0 \\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0} & 0&\underline{\mathbf{1}}&0&\underline{\mathbf{1}}&\underline{\mathbf{1}}&0&0 \\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0} & 0&0&\underline{\mathbf{1}}&0&\underline{\mathbf{1}}&0&\underline{\mathbf{1}} \\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0} & 0&0&0&\underline{\mathbf{1}}&\underline{\mathbf{1}}&\underline{\mathbf{1}}&0 \end{array}\right]$$

(4) The zero rows corresponds to the linearly dependent rows in Q as follows

$$eq_1 \Rightarrow q_1(\mathbf{x}) \oplus q_3(\mathbf{x}) \oplus q_4(\mathbf{x}) = x_2 \oplus x_3 \Rightarrow c_1 \oplus c_3 \oplus c_4 = m_2 \oplus m_3.$$

$$eq_2 \Rightarrow q_2(\mathbf{x}) \oplus q_4(\mathbf{x}) \oplus q_5(\mathbf{x}) = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \Rightarrow c_2 \oplus c_4 \oplus c_5 = m_1 \oplus m_2 \oplus m_3 \oplus m_4.$$

$$eq_3 \Rightarrow q_3(\mathbf{x}) \oplus q_5(\mathbf{x}) \oplus q_7(\mathbf{x}) = x_1 \Rightarrow c_3 \oplus c_5 \oplus c_7 = m_1.$$

$$eq_4 \Rightarrow q_4(\mathbf{x}) \oplus q_5(\mathbf{x}) \oplus q_6(\mathbf{x}) = x_2 \Rightarrow c_4 \oplus c_5 \oplus c_6 = m_2.$$

Given the known-ciphertext $\mathbf{c} = (c_1, \ldots, c_7) = (1101101)$, we solve the above linear equations to get plaintext $\mathbf{m} = (m_1, \ldots, m_4) = (0001)$. $m_5 = 1$ can be obtained by encrypting $(0001m_5^{'}), m_5^{'} \in \mathbb{F}_2$, using the known public key, and checking the corresponding ciphertext.

CHAPTER 4

Image Encryption Schemes Based on Multiple Parameters Transforms

Recent developments of generalized forms of signal processing transforms with a large number of independent parameters, such as the Multiple Parameter Fractional Fourier Transform and the Discrete Fractional Cosine Transform, have led many researchers to propose image encryption algorithms based on a single or multiple applications of these transforms. In order to claim a high level of security of these parameterized transforms-based schemes, their authors usually use the argument that the encrypted image is visually indistinguishable from random noise.

In this chapter, we show that these algorithms represent typical examples of insecure ciphers. All the building blocks of these schemes are linear, and hence, breaking these scheme, using a known plaintext attack, is equivalent to solving a set of linear equations. We also invalidate the argument of relying on the visual quality of the encrypted image ciphertext by presenting an example for a trivially insecure system that produces ciphertext images with the same property. An argument against the claimed efficiency of these schemes is also provided.

4.1 Introduction

Mathematical transforms, such as Fourier, Cosine, and Wavelets, have long been powerful tools for signal representation, analysis and processing. The discrete forms of these transforms have been widely applied to many domains including image processing.

Motivated by the wide available spectrum of possible applications, these transforms have been generalized. For example, the Fractional Fourier Transform (FRFT) [9] has been proposed as a generalization for the Fourier transform. The Fourier transform can be interpreted as a transform of a time domain signal into a frequency domain signal. Similarly, the interpretation of the inverse Fourier transform is as a transform of a frequency domain signal into a time domain signal. The FRFT, on the other hand, transforms a signal, either in the time domain or frequency domain, into the domain between time and frequency; it is a rotation in the time-frequency domain. The FRFT can be thought of as the Fourier transform to the n^{th} power, where it transforms a function to an intermediate domain between time and frequency. Its applications range from filter design and signal analysis to phase retrieval and pattern recognition. Unnikrishnan *et al.* [57] also proposed optical image encryption with FRFT.

With the increasing applications of FRFT, researchers have put numerous efforts on the development of its theory. Consequently, a variety of different forms of FRFTs were defined, which enriched the applications of these transforms. Zhu *et al.* [10] constructed a Multiple Fractional Fourier Transform (MFRFT) as a linear combination of the conventional FRFT. Afterward, Liu *et al.* [11] proposed the Random Fractional Fourier Transform (RFRFT) by randomizing the transform kernel function of the conventional FRFT. Later, Tao *et al.* [58] proposed the Multiple-Parameter Fractional Fourier Transform (MPFRFT) by using the fractionalized method from Shih [59]. Following ideas from the FRFT, a series of transforms, such as Fractional Cosine Transform (FCT) [12], Fractional Hadamard

Transform (FRHaT) [13], and Fractional Random Transform (FRT) [60], have been proposed.

A common feature of these generalized transforms is that they have a relatively larger number of independent parameters as compared to their corresponding original forms. Reconstructing the original signal from the transformed domain requires the application of the inverse transform with the exact set of parameters corresponding to the ones that were applied to the original signal. Any simple modification in these parameters would lead to the reconstruction of a completely distorted version of the signal. These observations have encouraged many researchers to propose image encryption algorithms based on a single or multiple steps of these transforms, where the parameters of these transforms are used as encryption keys. A quick review of the relevant signal processing literature would reveal a surprisingly very large number of image encryption schemes based on these transforms. For example, [61], [62] and [63] propose similar systems based on the Discrete Fractional Fourier Transform (DFRFT). To increase the number of the transform parameters, the authors in [64] introduced the Discrete Multiple Parameter Fractional Fourier Transform (DMPFRFT) and then proposed an image encryption algorithm based on it. Nu *et al.* [65] proposed an image encryption system based on the Mixed Discrete Fourier Transformation (MxDFT) in which the constructed transform matrix is represented as a linear combination of more than one DFRFT. To add more randomness to encrypted images whose energy concentrates around the corners or borders, the Random Discrete Fractional Fourier Transform (RDFRFT) was proposed in [15]. The eigenvectors of the kernel matrix of the RDFRFT are random DFT eigenvectors that are computed from eigenvectors of a random DFT-commuting matrix. Several discrete Fourier-related transforms have been parameterized in order to design image encryption algorithms. For example, Zhou *et al.* [66] present an image encryption system using the Discrete Parametric Cosine Transform (DPCT) and Tao *et al.* [13] present another system based on the Multiple-Parameter Discrete Fractional

Hadamard Transform (MPDFrHaT).

The rest of the chapter is organized as follows. In the next section, we briefly review the details of three representative samples of these image encryption schemes which are based on RDFRFT [15], Multi Orders Fractional Fourier Transforms [16] and Reciprocal-Orthogonal Parametric (ROP) Transform [17], respectively. In Section 4.3, we present our main observations about these schemes: all these schemes are linear and hence they can be trivially broken using a known plaintext attack. We also show that relying on the visual quality of the encrypted image does not provide any rigorous proof of security for the underlying system and we discuss the claimed efficiency of these schemes.

4.2 Examples of image encryption algorithms based on multiple parameters transforms

In this section, we briefly summarize three representative examples of the image encryption algorithms that are based on parameterized discrete transforms.

Image encryption based on RDFRFT [15]

Recall that the $N \times N$ DFT matrix is defined as

$$[\mathbf{F}]_{m,n} = \frac{1}{\sqrt{N}} e^{-j(2\pi/N)mn}, 0 \leq m, n \leq N-1. \tag{4.1}$$

The Eigen decomposition of the DFT matrix $\mathbf{F}$ is given by

$$\mathbf{F} = \sum_{k=0}^{N-1} \lambda_k \mathbf{e}_k \mathbf{e}_k^T, \tag{4.2}$$

where $\mathbf{e}_0, \mathbf{e}_1, \cdots, \mathbf{e}_{N-1}$ form an orthonormal eigenvector basis of the DFT.

The DFRFT $\mathbf{F}^a$ with one parameter a is defined as

$$\mathbf{F}^a = \sum_{k=0}^{N-1} \lambda_k^a \mathbf{e}_k \mathbf{e}_k^T. \tag{4.3}$$

A generalization of DFRFT, called the *Discrete Multiple Parameters Fractional Fourier Transform* (DMPFRFT) [64], is defined with the corresponding matrix

$$\mathbf{F}^{\bar{a}} = \sum_{k=0}^{N-1} \lambda_k^{a_k} \mathbf{e}_k \mathbf{e}_k^T, \tag{4.4}$$

where $\bar{a} = [a_0, a_1, \cdots, a_{N-1}]$.

The RDFRFT [15] with $1 \times N$ parameter vector $\bar{a}$ is obtained by using the random DFT-commuting matrix $\mathbf{H}$ and the DMPFRFT, as follows

$$\mathbf{F}_{\mathbf{H}}^{\bar{a}} = \sum_{k=0}^{N-1} \lambda_k^{a_k} \mathbf{r}_k \mathbf{r}_k^T, \tag{4.5}$$

where $\mathbf{r}_k$ are the orthonormal random DFT eigenvectors computed from $\mathbf{H}$, and λ_k is the DFT eigenvalue corresponding to $\mathbf{r}_k$. The reader is referred to [15] for the steps of obtaining matrix $\mathbf{H}$.

Finally, the 2-D RDFRFT image encryption with secret key parameters $(\bar{a}_1, \mathbf{H}_1, \bar{a}_2, \mathbf{H}_2)$ of an $N \times M$ image $\mathbf{P}$ is defined by

$$\mathbf{Q} = \mathbf{F}_{\mathbf{H}_1}^{\bar{a}_1} \cdot \mathbf{P} \cdot \mathbf{F}_{\mathbf{H}_2}^{\bar{a}_2}, \tag{4.6}$$

where $\mathbf{Q}$ is the encrypted image and $\mathbf{F}_{\mathbf{H}_1}^{\bar{a}_1}$, $\mathbf{F}_{\mathbf{H}_2}^{\bar{a}_2}$ are the $N \times N$ and $M \times M$ RDFRFT matrices, respectively.

Image encryption based on the multi-orders fractional Fourier transform [16]

The $N \times N$ of the p-th order DFRFT is defined as

$$\mathbf{F}_p^N = \mathbf{\Lambda}_{p,u}^N \cdot \mathbf{W}^N \cdot \mathbf{\Lambda}_{p,t}^N, \tag{4.7}$$

where the matrices $\mathbf{F}_p^N, \mathbf{W}^N \in \mathbb{C}^{N\times N}$ and the diagonal matrices $\mathbf{\Lambda}_{p,u}^N, \mathbf{\Lambda}_{p,t}^N \in \mathbb{C}^{N\times N}$ are defined as

$$[\mathbf{W}^N]_{m,n} = e^{-j\frac{2\pi}{N}(m-1)(n-1)},$$

$$[\mathbf{\Lambda}_{p,t}^N]_{n,n} = e^{j\frac{1}{2}\cot(\frac{p\pi}{2})(n-1)^2\cdot\Delta t^2},$$

and

$$[\mathbf{\Lambda}_{p,u}^N]_{n,n} = e^{j\frac{1}{2}\cot(\frac{p\pi}{2})(n-1)^2\cdot\Delta u_p^2},$$

where $p \in (0, 2)$, Δt is the sampling intervals in the time domain, and $\Delta u_p = 2\pi \sin(p\pi/2)/(N\Delta t)$ is the p-th order of Fractional Fourier domain (FRFD). Correspondingly, the p-th order of the inverse DFRFT for an N-length sequence, which is equivalent to the p-th order DFRFT, can be expressed as the Hermite transposition of $\mathbf{F}_p^N$, i.e.,

$$\mathbf{F}_{-p}^N = (\mathbf{F}_p^N)^H = \mathbf{\Lambda}_{-p,t}^N \cdot (\mathbf{W}^N)^H \cdot \mathbf{\Lambda}_{-p,u}^N. \tag{4.8}$$

The encryption is carried as follows: The $A \times B$ original image $\mathbf{P}$ is divided into $M \times N$ sub-images, $\mathbf{P}_{a,b}$, $a = 1, 2, \cdots, M$ and $b = 1, 2, \cdots, N$, with a size of $A/M \times B/N$, which are given by

$$[\mathbf{P}_{a,b}]_{m,n} = [\mathbf{P}]_{(a-1)A/M+m,(b-1)B/N+n}. \tag{4.9}$$

If the column vectors of $\mathbf{P}_{a,b}$ are M-fold interpolated and further $p_{a,b,a}$-th order inverse discrete fractional Fourier transformed, then we can obtain the encrypted sub-images $\mathbf{Q}_{a,b}$, based on the FRFD analysis of the interpolation [67, 68], as

$$\mathbf{Q}_{a,b} = \begin{bmatrix} \mathbf{D}_{p_{a,b,1},0} \\ \mathbf{D}_{p_{a,b,1},1} \\ \vdots \\ D_{p_{a,b,1},M-1} \end{bmatrix} \mathbf{X}_{a,b}[\mathbf{E}_{p_{a,b,2},0}\ \mathbf{E}_{p_{a,b,2},1}\ \cdots\ \mathbf{E}_{p_{a,b,2},N-1}], \tag{4.10}$$

where the matrix $\mathbf{X}_{a,b} \in \mathbb{C}^{(A/M)\times(B/N)}$ is the two dimensional inverse DFRFT of $\mathbf{P}_{a,b}$, which is given by

$$\mathbf{X}_{a,b} = \mathbf{F}^{A/M}_{-p_{a,b,1}} \cdot \mathbf{P}_{a,b} \cdot (\mathbf{F}^{B/N}_{-p_{a,b,2}})^T.$$

The diagonal matrices $\mathbf{D}_{p_{a,b,1},l} \in \mathbb{C}^{(A/M)\times(A/M)}$, $l = 0, 1, \cdots, M-1$ and $\mathbf{E}_{p_{a,b,2},k} \in \mathbb{C}^{(B/N)\times(B/N)}$, $k = 0, 1, \cdots, N-1$ are expressed as

$$[\mathbf{D}_{p_{a,b,1},l}]_{n,n} = e^{-j\frac{1}{2}\cot(\frac{p_{a,b,l}\pi}{2})\cdot[2(n-1)l(A/M)+l^2(A/M)^2]\Delta t^2}$$

$$[\mathbf{E}_{p_{a,b,2},k}]_{n,n} = e^{-j\frac{1}{2}\cot(\frac{p_{a,b,2}\pi}{2})\cdot[2(n-1)k(B/N)+k^2(B/N)^2]\Delta t^2}.$$

Then, we obtain the encrypted image $\mathbf{Q}$ by summation of $\mathbf{Q}_{a,b}$

$$\mathbf{Q} = \sum_{a=1}^{M}\sum_{b=1}^{N} \mathbf{Q}_{a,b}. \tag{4.11}$$

Image encryption based on the ROP transform [17]

Any integer n can be represented by r binary digits b_i, $0 \le i \le r-1$. If $(-1)^{\sum_{i=0}^{r-1} b_i} = -1$, then n is called a *minus integer*. The rows of any $N \times N$ matrix are indexed by integer numbers $0, 1, \cdots, N-1$. A row indexed by a minus integer is called a *minus-indexed row*.

The Reciprocal-Orthogonal Parametric (ROP) $N \times N$ matrix is constructed using a

parametric row vector $\mathbf{V}$ of length N and Hadamard matrix $\mathbf{H}$ of order N.

Throughout the rest of this section, we use a toy example with $N = 4$ to illustrate the basic principles of this scheme. The construction of the ROP transform matrix proceeds as follows,

- Form a parametric row vector $\mathbf{V} = [1\ a_1\ a_1\ 1]$, with the parameter a_1 is nonzero scalar and arbitrarily chosen from the complex plane.

- Construct the Hadamard matrix of order 4

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

which has four rows that can be indexed from the top to the bottom by 0, 1, 2, 3. In this case, the minus-indexed rows are those that are indexed by 1 and 2. By performing an element-by-element multiplication of each of these minus-indexed rows by the parametric vector $\mathbf{V}$ we obtain the normalized ROP matrix of order four

$$\mathbf{T}_4^{\mathbf{V}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -a_1 & a_1 & -1 \\ 1 & a_1 & -a_1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Let us denote by $\mathbf{T}_N^{\mathbf{V}_i}$ the ROP transform matrix of order *N* constructed using the parametric vector $\mathbf{V}_i$, which has $N/2 - 1$ independent parameters. Then, we consider four different parametric vectors $\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{V}_3$, and $\mathbf{V}_4$ to construct the ROP transform matrices $\mathbf{T}_N^{\mathbf{V}_1}$, $\mathbf{T}_N^{\mathbf{V}_2}$, $\mathbf{T}_N^{\mathbf{V}_3}$, and $\mathbf{T}_N^{\mathbf{V}_4}$, respectively.

The output of the first round is obtained as

$$\mathbf{C} = \mathbf{T}_N^{\mathbf{V}_2}(\mathbf{P} \odot [e^{j\alpha(n,m)}])\mathbf{T}_N^{\mathbf{V}_1}, \tag{4.12}$$

whereas the encrypted image $\mathbf{Q}$ of the original image $\mathbf{P}$ of size $N \times N$ is obtained as

$$\mathbf{Q} = \frac{1}{N^2}\mathbf{T}_N^{\mathbf{V}_4}(\mathbf{C} \odot [e^{j\beta(n,m)}])\mathbf{T}_N^{\mathbf{V}_3}, \tag{4.13}$$

where $\odot$ denotes the element-by-element multiplication operation of matrices, $[e^{j\alpha(n,m)}]$ and $[e^{j\beta(n,m)}]$ are two $N \times N$ random phase matrices, $\alpha(n,m)$ and $\beta(n,m)$, $1 \leq n, m \leq N$, are white, uniformly distributed in $[0, 2\pi]$ and independent of each other. The $(2N-4)$ independent parameters $(\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_4)$ of the matrices $\mathbf{T}_N^{\mathbf{V}_1}$, $\mathbf{T}_N^{\mathbf{V}_2}$, $\mathbf{T}_N^{\mathbf{V}_3}$, and $\mathbf{T}_N^{\mathbf{V}_4}$ and the random phase matrices are used as the encryption secret keys. Figures 4.1 and 4.2 show the encryption and decryption processes, respectively.

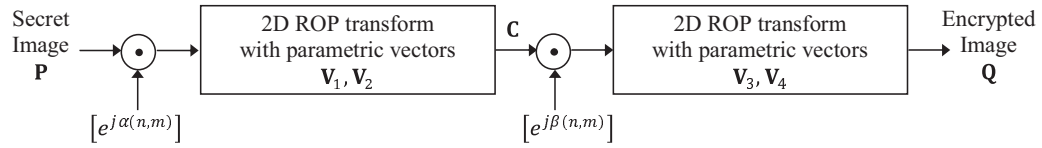


Figure 4.1: Encryption process based on ROP transform domain.

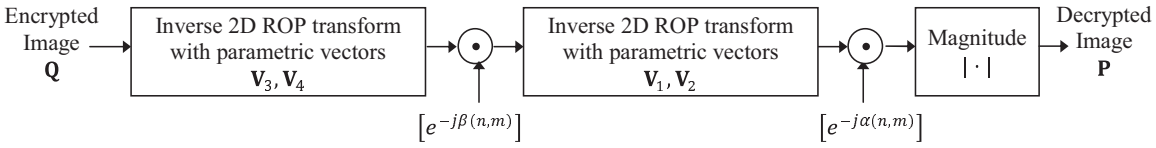


Figure 4.2: Decryption process based on ROP transform domain.

4.3 Main observations

An introduction to different types of cryptanalytic attacks can be found in [69]. A more rigorous mathematical treatment can be found in [1]. The attack described here is a known plaintext attack, i.e., we assume that the cryptanalyst can observe some of the plaintext images and its corresponding encrypted ciphertext. One should note that, because of the large size of the key required to encrypt an image using these multiple parameter transforms, the assumption that the same encryption key will be used to encrypt several images is realistic; otherwise, the user is better off using the theoretically secure one-time pad algorithm [1].

4.3.1 Known plaintext attack

Despite the apparent complexity of some of the above generalized transforms, a common feature in all of them is that there is an equivalent matrix description for each one of them.

In order to avoid unnecessary mathematical notation, consider the toy example of the RDFRFT image encryption system presented in the previous section with $N = 2$.

Let

$$\mathbf{F}_{\mathbf{H}_1}^{\bar{\alpha}_1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \mathbf{F}_{\mathbf{H}_2}^{\bar{\alpha}_2} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \text{ and } \mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}.$$

Then we have

$$\begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{21} & a_{12}b_{11} & a_{12}b_{21} \\ a_{11}b_{12} & a_{11}b_{22} & a_{12}b_{12} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{21} & a_{22}b_{11} & a_{22}b_{21} \\ a_{21}b_{12} & a_{21}b_{22} & a_{22}b_{12} & a_{22}b_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \end{bmatrix}.$$

By noting that the decomposition of linear systems is linear. Similar relation follow for the two other systems presented in Section 2. Also, adding an arbitrary large number of rounds would not add any nonlinearity to these schemes. Consider an $N \times M$ image $\mathbf{P}$. Let $\mathbf{I}_{NM\times 1}$ and $\mathbf{O}_{NM\times 1}$ denote the column-vectors obtained by concatenating the elements of the input plaintext matrix, $\mathbf{P}$, and the output ciphertext matrix, $\mathbf{Q}$, respectively. Then, for all the above encryption systems, we have

$$\mathbf{O} = \mathbf{K} \times \mathbf{I} \tag{4.14}$$

where $\mathbf{K}$ is an $NM \times NM$ equivalent key matrix whose elements can be recovered using $O(NM)$ known plaintext-ciphettext pairs. While the complexity of solving the above system of equations using Gaussian elimination is given by $O((NM)^3)$, other more advanced techniques can reduce this complexity to $O((NM)^{2.37})$. It should be noted that this complexity is much less than the usually claimed security level of these systems.

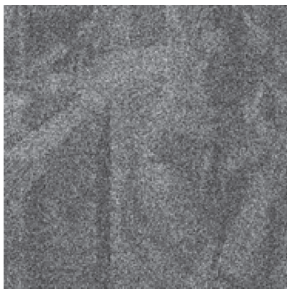
A folklore argument that is typically presented by the authors of image encryption schemes based on parameterized discrete transforms is that their proposed schemes are more secure than others because their underlying transforms utilize a larger number of transform parameters which increases the length of the corresponding secret keys. For example, the DMPFRFT-based scheme in [70] claim better security than DFRFT because it uses more parameters compared to the DFRFT.

Contrary to the above folklore argument, when considering this basic forms of known plaintext attacks, an encryption system based on a generalized transform with a huge number of parameters is equally as bad as any transform with very small number of parameters; both of them can be broken with the same complexity. Similarly, adding an arbitrary large number of transform rounds would not increase the security of these schemes.

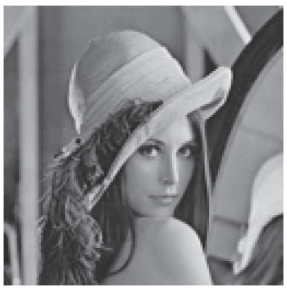
4.3.2 Visual quality of encrypted images

In order to claim a high level of security of these parameterized transforms-based schemes, their authors usually use the argument that the encrypted images are visually indistinguishable from random noise. While this is a necessary condition for any secure image encryption system, this condition is so loose to the extent that it can be satisfied by almost any system, even if it has a very poor security. Verifying the security of any image encryption scheme by visual observation is worthless practice; seeing a total random image as an encrypted image of the encryption scheme does not provide any assurance that the proposed algorithm is secure. As an illustration of this observation, Figure 4.4 shows the encrypted image using 11-bit Linear Feedback Shift register-based stream cipher (ciphertext only attack against this LFSR cipher, using the Berlekamp Massey algorithm [1], requires only 22 bits.) In fact, as depicted in Figure 4.3, this trivially insecure 11-bit LFSR shows better key avalanche properties when compared to some published schemes. It should be noted that the existence of such a high correlation between images decrypted with slightly incorrect keys and the original images may also facilitate ciphertext only attacks using heuristic search techniques.

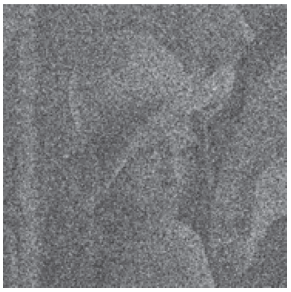
(a) Original Barbara



(b) Decrypted with incorrect key

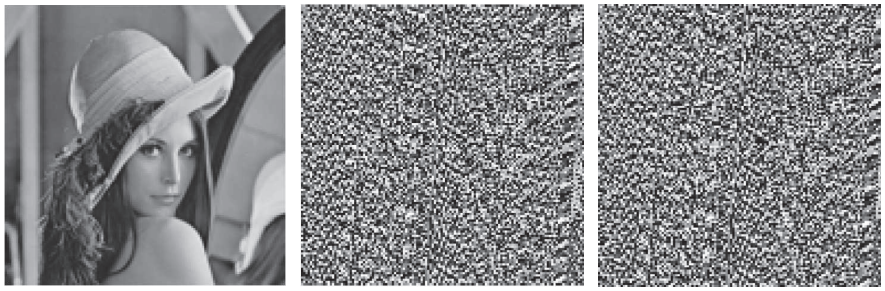


(c) Original Lena



(d) Decrypted with incorrect key

Figure 4.3: Examples of images decrypted with slightly incorrect keys for the ROP-based algorithm.



(a) (b) (c)

Figure 4.4: (a) Lena, (b) Lena encrypted by an LFSR, (c) Lena decrypted with a slightly incorrect key (1 bit difference).

4.3.3 Inefficiency of the proposed schemes

Due to the large data size and real-time constraints of multimedia data, some researchers argue that standard encryption algorithms may not be suitable for multimedia data. In fact, this reason is usually used as the main motivation for many parameterized transform based image encryption systems. However, since all the elements of the matrices corresponding to these parameterized transforms are complex numbers, the encryption process requires floating point operations which are much slower than the typical operations required by modern symmetric key ciphers. Also, there is usually a large data expansion associated with the encryption process because, unlike the plaintext, the ciphertext belongs to the set of complex numbers (which typically means an expansion by a $1 : 8$ factor.) Thus, current standard algorithms such as AES [3] outperform the above systems in terms of both encryption speed, bandwidth, and storage requirements.

CHAPTER 5

Secret Sharing Approaches for Images and 3D Objects

5.1 Introduction

The ongoing developments in computer technologies and the rapid increase in the number of Internet users have led to increasing the usage of network-based data transmission. In numerous applications, such as military documents and sensitive business data, this information must be kept secret and safe. Recently, 2D images and 3D models are considered as important as any other sensitive information. As a result, several 2D image-protection techniques, such as data encryption [71, 72] and steganography [73, 74], have been proposed to increase the security of 2D images. One common disadvantage of traditional protection techniques, such as encryption, is their policy of centralized storage. An entire protected model is usually maintained in a single information storage. If an intruder detects security vulnerability in the information storage in which the protected model resides, then s/he may attempt to decipher the secret model inside, or simply damage the entire information storage. Secret sharing is a defense mechanism that does not suffer from these problems. It works by splitting the secret into n shares that are transmitted and stored separately. One

can then reconstruct the original secret if at least a preset number t ($1 \leq t \leq n$) of these n shares are obtained. However, knowledge of less than t shares is insufficient for revealing the secret.

The idea of secret sharing was introduced independently in [18] and [75]. These schemes are based on the use of Lagrange interpolation polynomial and the intersection of affine hyperplanes, respectively. Since then, several studies have investigated the different implementations of the (t, n)-threshold scheme by mainly concentrating on secret sharing of keys in cipher systems. Most of these schemes are based on different mathematical primitives, such as matrix theory and prime numbers [1]. These protocols are specifically designed for text and numeric data. Due to the main distinctive nature of multimedia, in the sense that they have a large amount of data and the difference between two neighboring values is typically very small, it is considerably difficult to directly apply traditional secret sharing schemes to digital images or 3D objects. Thus, various secret sharing protocols have been designed exclusively for digital images based on vector quantization [76], Shamir-based schemes [77–79], sharing circle [80, 81], binary matrices [82], or cellular automata [83, 84]. Just a few of these schemes generate shares that have smaller sizes than the original image. The method proposed in [78] generates shares of $1/t$ the size of the secret image, whereas the method proposed in [77], that involves using Huffman coding scheme, generates shared images 40% smaller than that of the approach in [78].

Recently, flurry of research efforts has been carried out to design secure and efficient approaches for 2D image protection. However, 3D models have received less attention due to the fact that 2D image algorithms do not generally extend to 3D models. Besides, the rapid development in computer and information technology has increased the use of 3D models in various application domains, including manufacturing industries, entertainment and even in the military. Thus, the need for protection techniques to keep these 3D models secret and safe is of paramount importance. Inspired by the successful application of image

secret sharing schemes, in this chapter, we propose two secret sharing approaches for 3D models using Blakely scheme [75] and Thien & Lin [78]. We then show that encoding the 3D models using Huffman coding [85] or ZLIB [86] prior to secret sharing reduces the shares sizes significantly.

The rest of the chapter is organized as follows. In Section 5.2, we describe the image secret sharing scheme proposed in [82], and then we present the theoretical steps to recover the secret image from a single share only. We also provide experimental results on different images to validate the effectiveness of the recovering process. In Section 5.3, we review the 3D triangle mesh definition and the traditional secret sharing schemes. Then, we propose two secret sharing approaches for 3D models and describe their algorithmic steps. Finally, we apply the proposed approaches to various 3D models to demonstrate their effectiveness.

5.2 Matrix-based secret sharing scheme for images

Several attempts have been made to propose efficient secret sharing schemes for 2D images. The scheme proposed in [82] is a relatively fast image secret sharing scheme based on simple binary matrix operations. In this section, we show that care should be taken when choosing the matrices that corresponds to the shares. In particular if the rank of these singular matrices is not low enough then one can recover the secret image from only one share. Experimental results are provided to demonstrate the practicality of the recovery procedure on various 2D images.

5.2.1 $(2, n)$-threshold matrix-based image secret sharing scheme

The proposed secret sharing scheme in [82] is based on binary operations of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ that satisfy the following algebraic property. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two binary matrices such that $\boldsymbol{A} \bigoplus \boldsymbol{B} = \boldsymbol{Id}$. Then the following theorem holds

Theorem 5.1. $\boldsymbol{A}^m \oplus \boldsymbol{B}^m = \boldsymbol{I}$ *if and only if* $m = 2^e$, *with* $e \in Z^+$

The reader can refer to [82] for the proof of theorem 5.1.

The steps of the image secret sharing scheme proposed in [82] are described as follows,

The setup phase. Let the matrix $\boldsymbol{J} = (p_{i,j})$ be a gray-level image defined by $n \times n$ pixels such that $p_{i,j}$ is the numeric value of the gray color of the (i,j)-th pixel, where $p_{i,j} = (q^1_{i,j}, q^2_{i,j}, \cdots, q^8_{i,j}) \in \mathbb{Z}_2^8$, with $1 \leq i,j \leq n$. Consequently, eight binary matrices with coefficients in $\mathbb{Z}_2$ are extracted from $\boldsymbol{J} = (q^k_{i,j})$, where $1 \leq k \leq 8$. For simplicity, we write $\boldsymbol{J} = \boldsymbol{J}^1 \| \boldsymbol{J}^2 \| \ldots \| \boldsymbol{J}^8$ as a concatenation of eight binary matrices, where each of them represents a black and white subimage. Figures 5.1 and 5.2 correspond to Lena gray image defined by 128×128 pixels and its eight subimages, respectively.

The sharing phase. We choose two singular binary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ satisfying Theorem 5.1. Then, we randomly choose $n/2$ integer numbers $e_1 \leq e_2 \leq \ldots \leq e_{n/2}$ and computes $m_i = 2^{e_i}$. The shares $\mathbb{S}^1_i$, $\mathbb{S}^2_i$, where $1 \leq i \leq n/2$, are computed as follows,

$$\begin{aligned} \mathbb{S}^1_i &= \boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^1 \| \boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^2 \| \cdots \| \boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^8, \\ \mathbb{S}^2_i &= \boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^1 \| \boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^2 \| \cdots \| \boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^8. \end{aligned} \tag{5.1}$$

The recovery phase. The users P^1_i and P^2_i combine their shares, $\mathbb{S}^1_i$, $\mathbb{S}^2_i$, and compute the original image as follows,

$$\boldsymbol{J} = (\boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^1) \oplus (\boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^1) \| (\boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^2) \oplus (\boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^2) \| \cdots \| (\boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^8) \oplus (\boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^8).$$

The drawback of this scheme is that every participant P^1_i has only one qualified participant from the pool of participants that s/he can collude to recover the original image. If the share of the qualified participant P^2_i is altered or modified, the participant P^1_i will never be able to recover the original image.

To overcome this drawback, the author in [82] proposed a generalization of the above scheme, such that any participant P_i^1 can combine his share with any other participant P_j^2, $1 \leq i, j \leq n/2$ to recover the secret image. The generalization of the protocol is exactly the same as the basic one except for the sharing phase where the data $\{\mathbb{S}_i^1, m_i = 2^{e_i}\}$, $\{\mathbb{S}_i^2, m_i = 2^{e_i}\}$ are distributed to the participants P_i^1, P_i^2 respectively. Finally, in the recovery phase, the participants P_i^1, P_j^2 recover the secret $\boldsymbol{J}$ image as follows:

i) Compare the integer numbers m_i and m_j.

If $m_i < m_j$, P_i^1 computes

$$\boldsymbol{A}^{m_i+m_j-m_i} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{A}^{m_i+m_j-m_i} \cdot \boldsymbol{J}^8 = \boldsymbol{A}^{m_j} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{A}^{m_j} \cdot \boldsymbol{J}^8,$$

Otherwise, if $m_i > m_j$, P_j^2 computes

$$\boldsymbol{B}^{m_j+m_i-m_j} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{B}^{m_j+m_i-m_j} \cdot \boldsymbol{J}^8 = \boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^8.$$

ii) The recovery of the original image is carried as follows

$$\boldsymbol{J} = (\boldsymbol{A}^{m_j} \cdot \boldsymbol{J}^1) \oplus (\boldsymbol{B}^{m_j} \cdot \boldsymbol{J}^1) \| \cdots \| (\boldsymbol{A}^{m_j} \cdot \boldsymbol{J}^8) \oplus (\boldsymbol{B}^{m_j} \cdot \boldsymbol{J}^8)$$

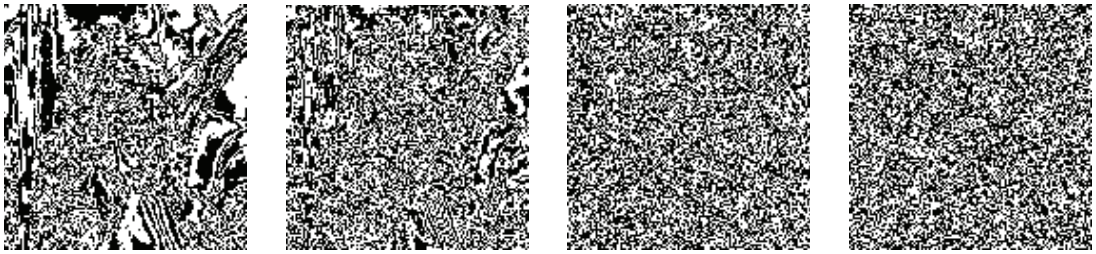
or

$$\boldsymbol{J} = (\boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^1) \oplus (\boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^1) \| \cdots \| (\boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^8) \oplus (\boldsymbol{B}^{m_i} \cdot \boldsymbol{J}^8)$$

We should highlight here that in order for the participants P_i^1 and P_j^2 to carry out the recovery phase of the generalized scheme, they must know the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

Figure 5.1: 128×128 Lena gray image.

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



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

Figure 5.2: (a) Subimage $\boldsymbol{J}^1$, (b) Subimage $\boldsymbol{J}^2$, (c) Subimage $\boldsymbol{J}^3$, (d) Subimage $\boldsymbol{J}^4$, (e) Subimage $\boldsymbol{J}^5$, (f) Subimage $\boldsymbol{J}^6$, (g) Subimage $\boldsymbol{J}^7$, (h) Subimage $\boldsymbol{J}^8$.

We should note that the most significant bits of the 8-bit values of grey images are the most important bits. One can distinguish the original image from the first subimages. For instance, as depicted in Figure 5.2 if we can recover $\boldsymbol{J}^1$ or $\boldsymbol{J}^2$ even partly we can recognize the original image. Secret sharing schemes based on the above idea should satisfy the following indistinguishability property: Given the secret share, one matrix should not be

able to distinguish this share from a random binary matrix of the same dimension, which is similar to the notion of semantic security in cryptography.

5.2.2 Recovering the original image from a single share

It is proven in ([87], Ch. A.3.3) that a real-valued matrix filled with independent and identically distributed random variables, with continuous probability distribution function, will be singular with probability zero. On the contrary, in $GF(2)$ the probability a square random binary matrix is singular as its dimension tends to infinity is 71.1% [88].

The scheme proposed in [82] stated that the random matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ must be singular without specifying the value of the ranks. Furthermore, generating a singular binary matrix randomly will produce a matrix that has high rank with high probability. Consequently, solving a set of linear equations when the coefficient matrix has a high rank will result in decreasing the number of independent/depended variables and increasing the number of determined variables.

The main weakness in the generalization of the image secret sharing scheme proposed in [82] is that for the participants to reconstruct the secret image they must know $\boldsymbol{A}$ and $\boldsymbol{B}$ along with the integer numbers m_i and m_j. For instance, the participants P_i^1 and P_j^2 must know ($\boldsymbol{A}$, $\boldsymbol{B}$, m_i, m_j) to perform the comparison between m_i and m_j and the computation $\boldsymbol{A}^{m_i+m_j-m_i} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{A}^{m_i+m_j-m_i} \cdot \boldsymbol{J}^8$ or $\boldsymbol{B}^{m_j+m_i-m_j} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{B}^{m_j+m_i-m_j} \cdot \boldsymbol{J}^8$, as stated in the generalization of the scheme.

To recover the original $n \times n$ gray image from a single share, the participant P_i^1 has his share $\mathbb{S}_i^1 = \boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^1 \| \cdots \| \boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^8$ and knows $\boldsymbol{A}$ and m_i. Then, he perform the following to recover $\boldsymbol{J}^1$,

Let $\boldsymbol{A} = (a_{i,j})$, $\boldsymbol{J}^1 = (J_{i,j})$ *and* $\boldsymbol{A}^{m_i} \cdot \boldsymbol{J}^1 = (s_{i,j})$. *Calculate* $\boldsymbol{A}^{m_i} = (\hat{a}_{i,j})$, $1 \leq i, j \leq n$

and form the set of linear Boolean equations as follows,

For $k = 1$ *to* n

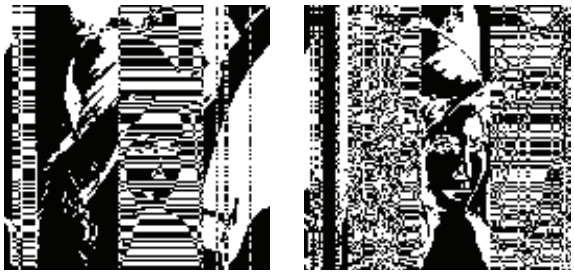
$$\begin{array}{lll} \bigoplus_{i=1}^{n} \hat{a}_{k,i} J_{i,1} & = & s_{k,1} \\ \bigoplus_{i=1}^{n} \hat{a}_{k,i} J_{i,2} & = & s_{k,2} \\ \vdots \quad\quad \vdots & & \vdots \quad . \\ \bigoplus_{i=1}^{n} \hat{a}_{k,i} J_{i,n-1} & = & s_{k,n-1} \\ \bigoplus_{i=1}^{n} \hat{a}_{k,i} J_{i,n} & = & s_{k,n} \end{array} \tag{5.2}$$

We have $n \times n$ binary linear equations with $n \times n$ variables. Solving this set of linear equations will produce determined variables (pixels) with specific values, depended variables, and independent variables which we can assign any value to them. Due to the nature of the image, there is no need to find the exact values of all the pixels. With only small percentage of the correct values of the pixels, we can distinguish the original image, as we can see from the recovered $\boldsymbol{J}^1$ and $\boldsymbol{J}^2$ in Figure 5.3. To recover the original gray image, we perform the previous recovery steps to all eight subimages and then concatenate them together.

5.2.3 Experimental results

We applied the recovery technique on different 128×128 gray images, namely, Lena, F16 and Fishing boat. The singular binary matrix $\boldsymbol{A}$ was chosen randomly with a rank equals 127 and the integer number m_i equals 2. We used the BooleanPolynomial class from the SAGE [89] package to do the fast calculations for solving the set of linear Boolean equations depending on 128×128 binary variables. As an example in the case of Lena, after solving the set of linear binary equations, the number of undetermined variables was

12928, which were set to zeros. On the other hand, the number of determined variables for each of $\boldsymbol{J}^1, \boldsymbol{J}^2 \cdots, \boldsymbol{J}^8$ was 3456, which is the number of bits (pixels) that were recovered exactly. Although this number seems too small comparing to 128×128 bits, it is enough to recognize the original image easily as shown in Figure 5.3.



(a) (b)

Figure 5.3: The recovered subimages of Lena (a) $\boldsymbol{J}^1$, (b) $\boldsymbol{J}^2$.

One can see from Figures 5.4 and 5.5 the similarity between the recovered images and the corresponding original images.

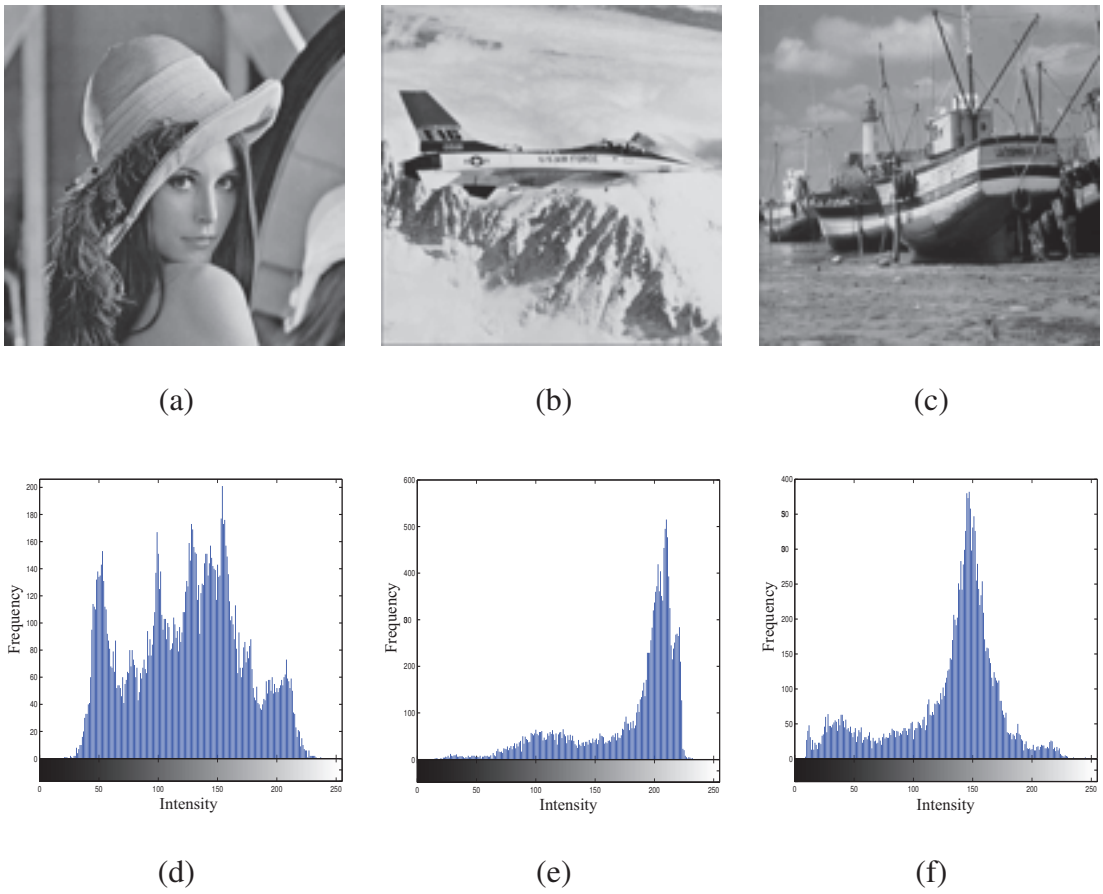


Figure 5.4: Original images of (a) Lena, (b) F16, (c) Fishing boat, (d)-(f) their histograms.

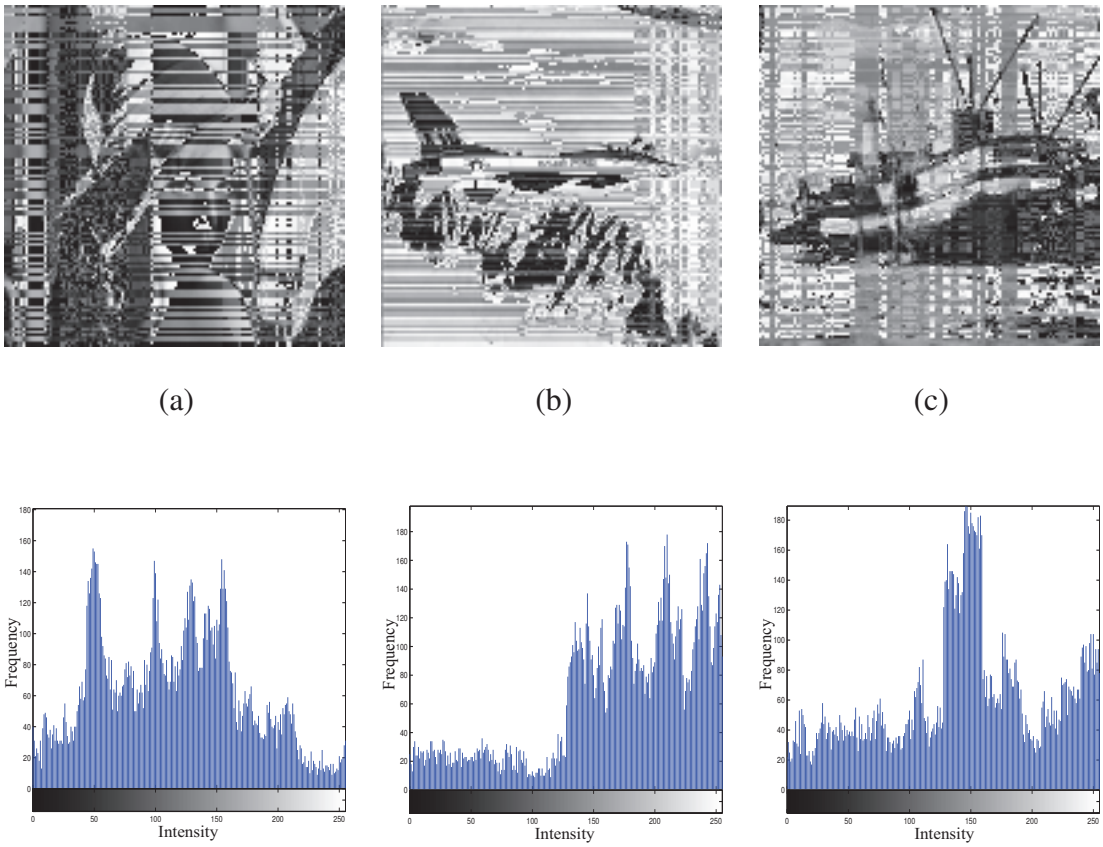


Figure 5.5: Recovered images of (a) Lena, (b) F16, (c) Fishing boat, (d)-(f) their histograms.

5.3 Secret sharing approaches for 3D objects

Inspired by the successful application of image secret sharing schemes in multimedia protection, in this section we present two secret sharing approaches for 3D models using Blakely and Thien & Lin schemes. We show that encoding 3D models using lossless data compression algorithms prior to secret sharing helps reduce share sizes and remove redundancies and patterns that possibly ease cryptanalysis. The proposed approaches provide a higher tolerance against data corruption/ loss than existing 3D protection mechanisms, such

as encryption. Experimental results are provided to demonstrate the secrecy and safety of the proposed schemes. The feasibility of the proposed algorithms is demonstrated on various 3D models.

5.3.1 Problem formulation

In computer graphics and geometric-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is defined as $\mathbb{M} = (\mathcal{V}, \mathcal{T})$ where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices and $\mathcal{T} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_\ell\}$ is the set of triangles (faces) as shown in Figure 5.6. In matrix form, the sets $\mathcal{V}$ and $\mathcal{T}$ may be written as follows:

$$\mathcal{V} = \begin{pmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_m \end{pmatrix} = \begin{pmatrix} v_{1_x} & v_{1_y} & v_{1_z} \\ v_{2_x} & v_{2_y} & v_{2_z} \\ \vdots & \vdots & \vdots \\ v_{m_x} & v_{m_y} & v_{m_z} \end{pmatrix}, \qquad \mathcal{T} = \begin{pmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \vdots \\ \boldsymbol{t}_\ell \end{pmatrix} = \begin{pmatrix} t_{1_i} & t_{1_j} & t_{1_k} \\ t_{2_i} & t_{2_j} & t_{2_k} \\ \vdots & \vdots & \vdots \\ t_{\ell_i} & t_{\ell_j} & t_{\ell_k} \end{pmatrix}$$

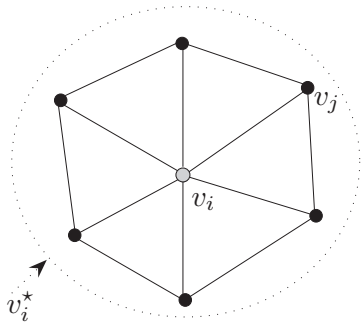


Figure 5.6: Vertex neighborhood $\boldsymbol{v}_i^\star$.

Secret sharing scheme is a method for distributing a secret amongst a group of participants, each of which is given a share of the secret. The secret can only be reconstructed

when the shares are combined together; individual shares are of no use on their own.

Blakley (t, n)-threshold scheme

The Blakley scheme uses hyperplane geometry to solve the secret sharing problem. The secret is a point in a t-dimensional space. The n shares are constructed such that each share is defined as an affine hyperplane that passes through the secret point. An affine hyperplane can be described by a linear equation of the following form $a_1x_1 + \ldots + a_tx_t = b$. The intersection point is obtained by finding the intersection of any t of these hyperplanes. The secret can be any of the coordinates of the intersection point or any function of the coordinates.

For example, for a $(3, n)$-threshold scheme:

1. Choose a prime number p larger than the point coordinates.

2. Given a secret point (x_0, y_0, z_0), n shares are generated as follows:

   For each share:

   2.1. Choose $a, b \in \mathbb{F}_p$ independently at random.

   2.2. Let

   $$c = z_0 - ax_0 - by_0 \mod p, \tag{5.3}$$

   where $z = ax + by + c$ is the equation of a hyperplane.

Given any t hyperplanes, the secret point is the intersection point of these t hyperplanes. Figure 5.7 illustrates how the three hyperplanes intersect in only one point (secret point).

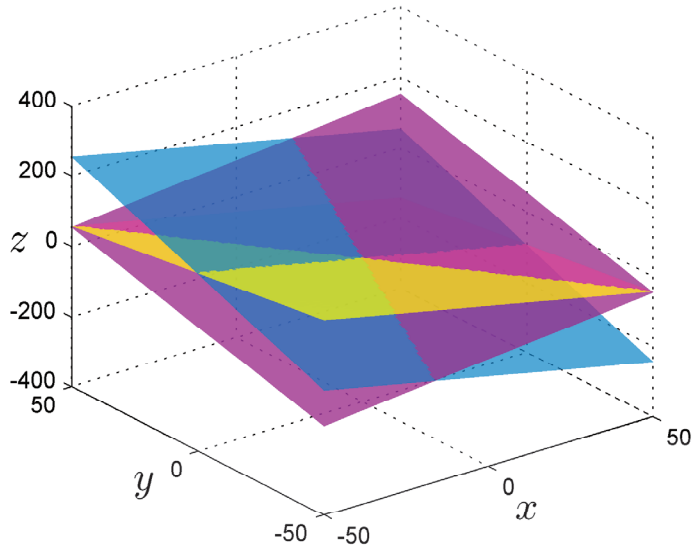


Figure 5.7: The secret point is the intersection point between the three planes.

The traditional Shamir and Blakley (t, n)-threshold schemes produce shares with same size as the original secret. In multimedia, a secret can be an image or 3D model. Typically, these files have a large amount of data. Thus, applying these traditional schemes may be inefficient in terms of the storage space. Consequently, several image secret sharing approaches have been proposed to reduce the shares sizes. The method proposed in [78] produces shares of $1/t$ the size of the secret image, whereas the Huffman coding-based scheme introduced in [77] generates shared images 40% smaller than the method in [78].

Thien & Lin (t, n)-threshold scheme

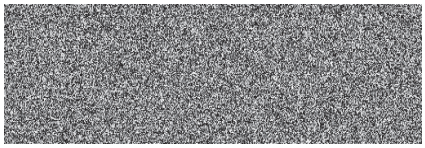
Thien & Lin proposed a (t, n)-threshold-based approach using Shamir scheme [18] for grayscale images to generate image shares. Suppose we want to divide the image S into n image shares $(S_1, \ldots, S_n)$, and the secret image S cannot be revealed without t or more image shares. The essential idea is to use a polynomial of degree $(t - 1)$ to construct n image shares, by letting the t coefficients be the gray values of t pixels. The main difference

is that these coefficients are randomly chosen in Shamir scheme. To this end, we first divide an image into several sections. Each section has t pixels of the image, and for each section j, the following $t-1$ degree polynomial is defined

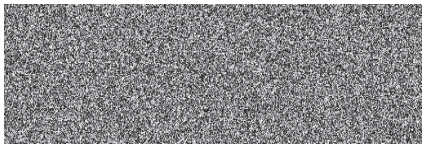
$$q_j(\mathbf{x}) = (a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}) \bmod p, \tag{5.4}$$

where the coefficients $a_0, a_1, \ldots, a_{t-1}$ are the values of the t pixels of the section. Then $q_j(1)$, $q_j(2)$, ..., $q_j(n)$ are computed. These n values of the section are distributed to the n participants to assign them sequentially to their n image shares. Since for each given section (of t entries) of the secret image, each image share receives only one value of the generated shares, the size of each image share is $1/t$ of the secret image. This method reduces the size of image shares to become $1/t$ of the size of the secret image. We should note here that since the gray values are between 0 and 255, the value of p was set to 251, which is the greatest prime number less than 255. For this method to be valid, all the pixel values greater than 250 must be rounded down to 250. Obviously, there will be some loss in terms of pixel values during the reconstruction of the secret image. Thus, Thien & Lin modified their technique to offer a lossless image secret sharing method. It should be noted here that applying Thien & Lin's scheme directly to the image shares can outline partially the original secret image. Therefore, some sort of initial permutation is needed before employing the scheme. Figure 5.8 illustrates Thien & Lin's image secret sharing scheme, where $t = 2$ and $n = 4$.

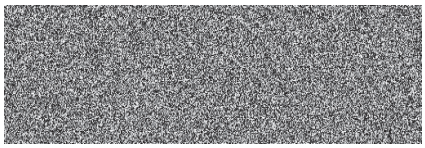
(a)



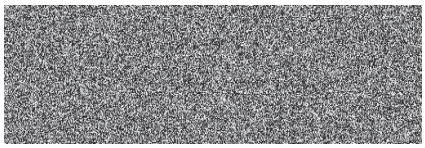
(b)



(c)



(d)



(e)

Figure 5.8: Thien & Lin's secret sharing process for a Jet plane: (a) original image 512×512, (b)-(e) the four share images after the original image is permuted, each of size $1/2$ of the original image size.

5.3.2 Proposed 3D secret sharing schemes

Motivated by the successful application of image secret sharing schemes, we propose in this section a secret sharing approaches for 3D models using Blakely scheme [75] and Thien & Lin [78]. Later, we show that encoding the 3D models using Huffman coding [85] or ZLIB [86] prior to splitting the secret reduces the shares sizes considerably.

Algorithm: Proposed 3D secret sharing scheme

For the faces matrix $\mathcal{T}$:

1. We choose $p_{\mathcal{T}}$ as the next prime number larger than m.
2. For each i-th share, $1 \leq i \leq n$:

   (i) Select two random numbers independently $a_i, b_i \in \mathbb{F}_{p_{\mathcal{T}}}$.

   (ii) Find c_i using Eq.(5.3), where x, y, z are the values of the face and a_i, b_i, c_i are the coefficients of the hyperplane equation $z = a_i x + b_i y + c_i$.

   (iii) Distribute the hyperplane equations (coefficients) to all n participants.
3. Repeat step (2) for all ℓ faces in $\mathcal{T}$ matrix.

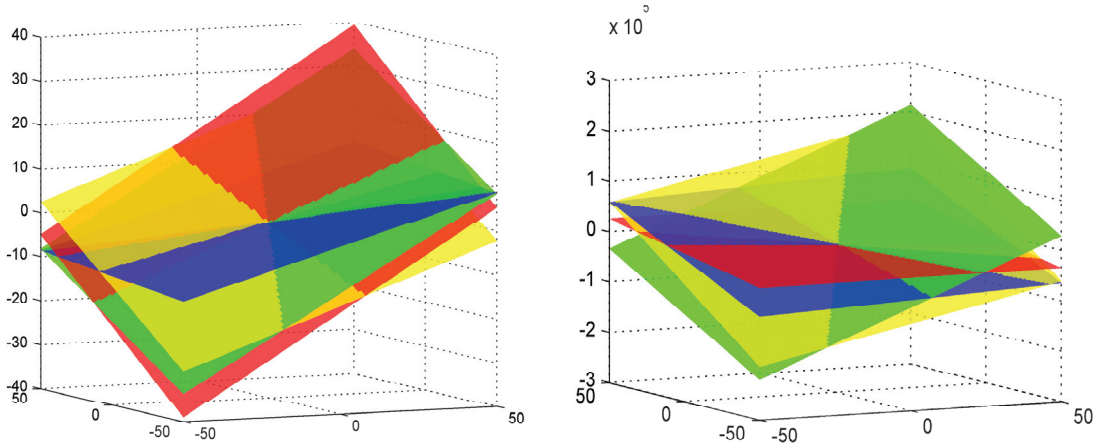
Table 5.1: Algorithmic steps of the proposed approach.

3D sharing secret using Blakley scheme

Our proposed approach is motivated by Blakley secret sharing scheme. The main idea is to split every vertex in the vertices matrix $\mathcal{V}$ and every face in the faces matrix $\mathcal{T}$ into n share hyperplanes, where $n > 3$. Each share hyperplane is represented by an equation $z = ax+by+c$. The main algorithmic steps of the proposed scheme are shown in Table 5.1.

In the recovery phase, the original values of the faces and vertices coordinates are the intersection points between any three or more hyperplane equations. To be able to draw the shares, all the calculations are performed in a prime field $\mathbb{F}_{p_{\mathcal{T}}}$. This $\mathbb{F}_{p_{\mathcal{T}}}$ is essential in splitting the faces matrix $\mathcal{T}$ to ensure the coefficients of the shared hyperplanes are within the range of the number of vertices, i.e. less than m. Moreover, since all the values in $\mathcal{T}$ are integers, $\mathbb{F}_{p_{\mathcal{T}}}$ is necessary to avoid the prediction of the range of the faces original-values from the shared-values. On the other hand, the modular operation in splitting the

vertices matrix is not crucial. In this case, the scheme is still secure since the vertices coordinates (v_x, v_y, v_z) and the random coefficients (a_i, b_i) are floating number, negative numbers, and sometimes integers. For this reason, the values of the shares will not correlate to the original-values of the vertices matrix. Thus, we apply the same algorithmic steps to split the vertices matrix $\mathcal{V}$, with the exception of using the prime field. Knowing the numbers of the vertices and faces of the share, the adversary can guess the share corresponds to which 3D model. To resist this statistical attack, we duplicate the last vertex (resp. last face) to a random numbers prior to finding $p_{\mathcal{T}}$. Figure 5.9 shows how Blakley scheme split the vertex $\boldsymbol{v}_1$ and face $\boldsymbol{t}_1$ of the 3D F15 fighter jet model into four share hyperplanes.



(a) (b)

Figure 5.9: Four planes generated by Blakley secret sharing scheme of F15 model for (a) vertex $\boldsymbol{v}_1$, and (b) face $\boldsymbol{t}_1$.

3D sharing secret using Thian & Lin scheme

Blakley's scheme produces shares of the same size as the original (secret) 3D model. However, Thian & Lin scheme produces shares $1/3$ the size of the secret model. In the latter scheme, we divide each vertex (resp. each face) into n shares, where each share is $m \times 1$

array (resp. $\ell \times 1$ array), as shown in Figure 5.10.

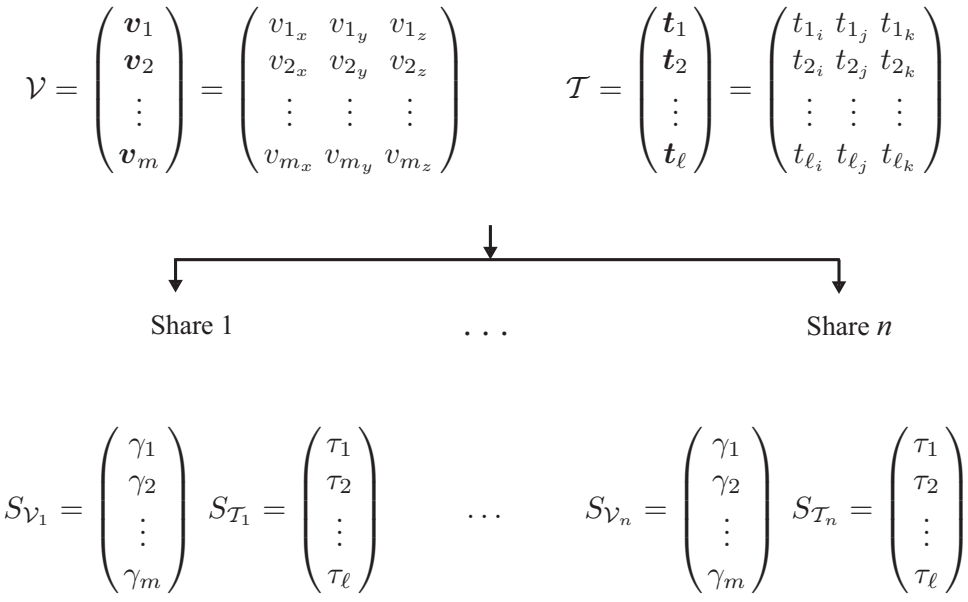


Figure 5.10: Thian & Lin secret sharing process: each share has two sub-shares $m \times 1$ vertices array and $\ell \times 1$ faces array.

To split the faces matrix $\mathcal{T}$, we use the vertex coordinates $\boldsymbol{v}_i = \{v_{i_x}, v_{i_y}, v_{i_z}\}$, $1 \leq i \leq m$, and the face values $\boldsymbol{t}_e = \{t_{e_i}, t_{e_j}, t_{e_k}\}$, $1 \leq e \leq l$, as the coefficients to the Eq.(5.4), where $t = 3$. The main difference between Shamir's scheme and Thian & Lin scheme is that the coefficients are not taken randomly. An important issue in the implementation of secret sharing schemes is the size of the shares, since the security of a system lessens as the amount of the information that must be kept secret increases. Unfortunately, in most secret sharing schemes the size of the shares cannot be less than the size of the secret. Therefore, to reduce the share size, we compress the 3D models using lossless data compression methods such as Huffman coding [85] or ZLIB [86] before applying the secret sharing schemes. Besides, compression prior to secret sharing helps remove redundancies and patterns that might facilitate cryptanalysis. ZLIB is lossless data compression library that uses a compression algorithm called *Deflate*. This lossless data compression algorithm uses a

3D Object	Vertices	Faces	Size (in bytes)
F15	5401	9665	176.55 KB
Tank	8659	18474	317.96 KB
Engine	2378	4562	81.32 KB

Table 5.2: Sizes of different 3D objects in bytes (Single precession)

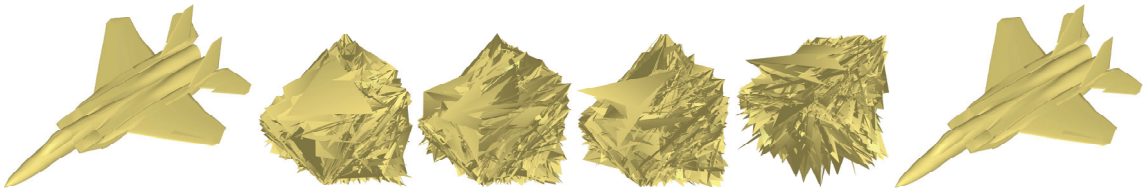
3D Model	Uncompressed	Compressed Using Huffman Code		Compressed Using ZLIB	
F15	176.55 KB	150.56 KB	85.2 %	98.38 KB	55.7 %
Tank	317.96 KB	271.15 KB	85.2 %	143.90 KB	45.2 %
Engine	81.32 KB	66.88 KB	82.2 %	23.12 KB	28.4 %

Table 5.3: Compression results of 3D objects using Huffman coding and ZLIB algorithm.

combination of LZ77 algorithm and Huffman coding, and provides good compression on a wide variety of data with minimal use of system resources. Tables 5.2 and 5.3 show the original and compressed sizes of different 3D models.

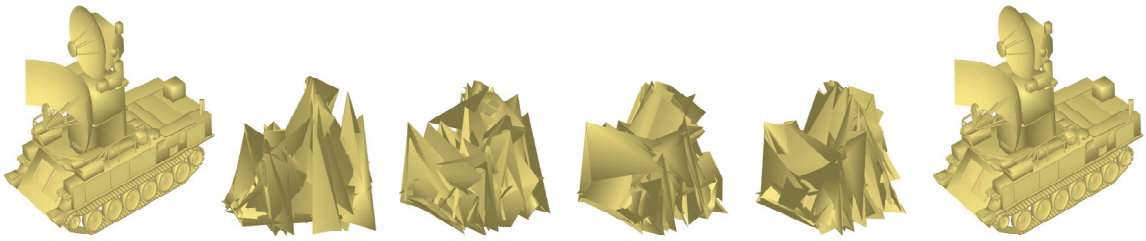
5.3.3 Experimental results

We applied the $(3, 4)$-Blakley scheme on two 3D models: F15 fighter jet and tank. The 3D F15 model consists of 5401 vertices and 9665 faces, whereas the 3D tank model consists of 8659 vertices and 18474 faces. All the share models have the same number of vertices and faces. If the number of vertices is not a prime number, then we may duplicate the last vertex until we reach the next prime number greater than the number of vertices. For F15 and tank models, we used the prime numbers $p_{\mathcal{T}} = 5407$ and $p_{\mathcal{T}} = 8663$, respectively. From Figures 5.11 and 5.12, it is clear that the four generated shares of both models are unrecognizable, indicating that the secret property is satisfied. Combining any 3 shares from Figures 5.11(b)-(e) (resp. Figures 5.12(b)-(e)), we can reconstruct the original secret model as shown in Figure 5.11(f) (resp. Figure 5.12(f)). Therefore, the lost of one share



(a) (b) (c) (d) (e) (f)

Figure 5.11: $(3, 4)$-Blakley secret sharing process for the 3D F15 model: (a) original model, (b)-(e) the four split shares, (f) reconstructed model using any 3 shares.



(a) (b) (c) (d) (e) (f)

Figure 5.12: $(3, 4)$-Blakley secret sharing process for the 3D tank model: (a) original model, (b)-(e) the four split shares, (f) reconstructed model using any 3 shares.

will not prevent recovery of the model. These results are in fact consistent with numerous 3D models used for experimentation.

To further decrease the amount of the information that must be kept secret and reduce the overhead calculations of the sharing process, we compress the 3D models before applying the secret sharing schemes. Table 5.4 shows the comparison between the sizes of the shares generated by Blakely and Thian & Lin schemes using Huffman coding [85] or ZLIB [86].

3D Model	**Blakely Scheme Share Size**			**Thian & Lin Scheme Share Size**		
	Uncompressed	**Huffman**	**ZLIB**	**Uncompressed**	**Huffman**	**ZLIB**
F15	176.55 KB	150.56 KB	98.38 KB	58.85 KB	50.18 KB	32.79 KB
Tank	317.96 KB	271.15 KB	143.90 KB	105.98 KB	90.38 KB	47.96 KB
Engine	81.32 KB	66.88 KB	23.12 KB	27.1 KB	22.29 KB	7.7 KB

Table 5.4: Comparison between the sizes of the shares generated by Blakely and Thian & Lin schemes using using Huffman coding and ZLIB compression algorithms.

CHAPTER 6

Conclusions and Future Research Directions

This chapter briefly concludes the thesis and highlights the major contributions of this research.

This thesis systematically studied various aspects of cryptography starting from the primitive components (such as Boolean functions) to cryptographic applications that deliver a secure multimedia contents.

In the next section, the contributions made in each of the previous chapters and the concluding results drawn from the associated research work are presented. Suggestions for future research directions related to this thesis are provided in Section 6.2.

6.1 Thesis contributions

6.1.1 Properties of cryptographic functions

In chapter 2, we first introduced the concept of nonlinearity-profile for cryptographic Boolean function. This new measure can be used to evaluate the strength of a Boolean function when we fix a subset of its input coordinates. The results in this section can be extended to other

cryptographic criteria. In the second part of the chapter, we investigated the existence of linear structures in rotation symmetric Boolean functions. The obtained results can help the researchers to identify and choose appropriate functions to be used in their desired cryptographic algorithms. At the end of the chapter, we generalized some of the previous results on cryptographic Boolean functions to functions defined over $GF(p)$. Finally, we presented a method to construct bent functions from semi-bent functions over $GF(p)$.

6.1.2 Cryptanalysis of a public key cryptosystems based on Boolean permutations

In chapter 3, we showed the public key cryptosystem (PKC2) proposed by Wu and Varadharajan [53] is insecure. In particular, when a linear function is used to initialize the algorithm used to construct the Boolean permutation employed in this system, the cryptanalyst can recover the plaintext from the corresponding ciphertext by solving a set of equations over $GF(2)$ using the known public key.

6.1.3 Image encryption schemes based on parameterized discrete transforms

In chapter 4, we showed that previously proposed image encryption algorithms based on discrete transforms represent typical examples of insecure ciphers due to their inherent linearity. A careful analysis of the performance of these algorithms also reveals their inefficiency in terms of both bandwidth and throughput. For practical applications requiring block ciphers, we recommend the use of the AES algorithm. Similarly, for applications requiring stream ciphers, we recommend the use of one of the seven stream ciphers recommended by the European ECRYPT stream cipher project, in the eSTREAM Portfolio. These algorithms have undergone extensive cryptanalytic reviews by the cryptographic

community and are optimized to achieve an excellent tradeoff between security and performance.

6.1.4 Secret sharing schemes for images and 3D models

In chapter 5, we showed that the matrix-based secret sharing scheme for 2D images proposed by by M. Rey [82] is not ideal for practical usage. Any participant in the scheme can recover the original image without the need to combine his share with any other participant. We also proposed in the chapter a geometric framework for 3D secret sharing. The proposed algorithms were motivated by Blakley and Thien and Lin secret sharing schemes. To increase the security of the schemes by decreasing the amount of the information that must be kept secret, we used two lossless data compression algorithms. Our experimental results on several 3D models indicate the feasibility of the proposed approaches.

6.2 Future research directions

Several interesting research directions motivated by this thesis are discussed next. In addition to constructing and analyzing the cryptographic Boolean functions and their generalization over different finite fields, we intend to accomplish the following projects in the near future:

6.2.1 Cryptographic Boolean functions

Exploring the applications of off-the-shelf SAT solvers as tools to answer some of the interesting open problems in designing Boolean functions. For instance, the construction of $(8, -, -, 118)$ boolean functions and studying the nonexistence of homogeneous bent RSBFs conjectured in [6].

6.2.2 Image encryption

Another future work direction is to investigate the use of a self-adaptive search optimization techniques such as Particle Swarm Optimization (PSO) in automated cryptanalysis of image encryption schemes based on parameterized discrete transforms. The preliminarily results showed that fine tuning the sensitive parameters of the PSO algorithm can partly reveal the original images. It is also interesting to further explore partial or selective encryption mechanisms for digital images and 3D models as they reduce the amount of data to encrypt while preserving a sufficient level of security.

6.2.3 Secret sharing schemes for 3D models

Another possible future research direction is the use of mesh compression techniques such as spectral mesh compression for 3D secret sharing. The key idea is to apply eigen-decomposition to the mesh matrix, and then discard the largest eigenvalues/eigenvectors in order to reduce the dimensionality of the new spectral basis so that most of the energy is concentrated in the low frequency coefficients. The advantage of incorporating mesh compression techniques for 3D secret sharing is mainly due to the large reduction of the resulting 3D model shares.

List of References

[1] A. Menezes, P. Van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton 1997.

[2] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *ACM Communications*, vol. 21, no. 2, pp. 120-126, 1978.

[3] FIPS 197. Advanced Encryption Standard. Federal Information Processing Standards Publication 197, U.S. Department of Commerce/N.I.S.T, 2001.

[4] C. Carlet, *Vectorial Boolean Functions for Cryptography,* Chapter of the monography Boolean Methods and Models, Y. Crama and P. Hammer eds, Cambridge University Press, 2007.

[5] S. Kavut, S. Maitra, and M. D. Ycel, "Search for Boolean functions with excellent profiles in the rotation symmetric class," *IEEE Trans. on Information Theory*, vol. 53, no. 5, pp. 1743-1751, 2007.

[6] P. Stanica and S. Maitra, "Rotation symmetric Boolean functions - count and cryptographic properties," *Bose Centenary Symposium on Discrete Mathematics and Applications, Electronic Notes in Discrete Mathematics*, vol. 15, pp. 139-145, 2002.

[7] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory,* vol. 31, no. 4, pp. 469-472., 1985.

[8] C. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, pp. 656-715, 1948.

[9] H. Ozaktas and D. Mendlovic, “Fractional Fourier transforms and their optical implementation,” *Journal of the Optical Society of America A: Optics and Image Science and Vision*. vol. 10, no. 12, pp. 2522-2531, 1993.

[10] B. Zhu, S. Liu and Q. Ran, “Optical image encryption based on multifractional Fourier transforms,” *Optics Letters*, vol. 25, pp. 1159-1161, 2000.

[11] Z. Liu and S. Liu, “Random fractional Fourier transform,” *Optics Letters*, vol. 32, pp. 2088-2090, 2007.

[12] S. Pei and M. Yeh, “The discrete fractional cosine and sine transforms,” *IEEE Trans. on Signal Processing*, vol. 49, no. 6, pp. 1198-1207, 2001.

[13] R. Tao, J. Lang and Y. Wang,“ The multiple-parameter discrete fractional Hadamard transform,” *Optics Communications*, vol. 282, no. 8, pp. 1531-1535, 2009.

[14] J. Guo, Z. Liu and S. Liu, “Watermarking based on discrete fractional random transform,” *Optics Communications*, vol. 272, pp. 344348, 2007.

[15] S. Pei and W. Hsue, “Random discrete fractional Fourier transform,” *IEEE Signal Process. Letters*, vol. 16, no. 12, 2009.

[16] R. Tao, X. Meng and Y. Wang, “Image encryption with multi-orders fractional Fourier transforms,” *IEEE Trans. on Information Forensics and Security*, vol. 5, no. 4, pp. 734-738, 2010.

[17] S. Bouguezel, A. Omair and M.N.S. Swamy, "Image encryption using the reciprocal-orthogonal parametric transform," *Proc. IEEE Symposium on Circuits and Systems ISCAS*, pp. 2542-2545, 2010.

[18] A. Shamir, "How to share a secret," *ACM Communications*, vol. 22, no. 11, pp. 612-613, 1979.

[19] Q. Wang, J. Peng, H. Kan and X. Xue, "Constructions of cryptographically significant Boolean functions using primitive polynomials," *IEEE Trans. on Information Theory*, vol. 56, no. 6, pp. 3048-3053, 2010.

[20] W. Zhang and G. Xiao, "Constructions of almost optimal resilient Boolean functions on large even number of variables," *IEEE Trans. on Information Theory*, vol. 55, no. 12, pp. 5822-5831, 2009.

[21] Y. Du, "Cryptographic properties of a class of Boolean functions with maximum algebraic immunity," *Proc. IEEE Conference on Computer Science and Information Technology ICCSIT*, pp. 612-615, 2010.

[22] J. Pieprzyk, T. Hardjono and J. Seberry, *Fundamentals of Computer Security*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[23] W. Millan, "Analysis and design of Boolean functions for cryptographic applications," PhD. Thesis, Queensland University of Technology, Australia, 1997.

[24] O. Rothaus, "On bent functions," *Journal of Combinatorial Theory*, Ser. A 20, pp. 300-305, 1976.

[25] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[26] M. Qing-Shu, Y. Min, Z. Huan-Quo and L. Yu-Zhen, "Analysis of affinely equivalent Boolean functions," Cryptology ePrint Archive, Report 025/2005.

[27] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. on Computers*, vol. C-34, no. 1, pp. 81-84, 1985.

[28] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Proc. CRYPTO - Lecture Notes in Computer Science*, vol. 537, pp. 2-21, 1991.

[29] M. Matsui, "Linear cryptanalysis method for DES cipher," *Proc. EUROCRYPT - Lecture Notes in Computer Science*, vol. 65, pp. 386-397, 1994.

[30] A. Kipnis and A. Shamir, "Cryptanalysis of the HFE public key cryptosystem by relinearization," *Proc. Crypto - Lecture Notes in Computer Science*, vol. 1666, pp. 19-30, 1999.

[31] C. Carlet, "Hyperbent functions," *Proc. PRAGOCRYPT*, pp. 145-155, 1996.

[32] J. Evertse, "Linear structures in block ciphers," *Proc. Eurocrypt - Lecture Notes in Computer Science*, vol. 304, pp. 249-266, 1988.

[33] M. Hellman, R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig and P. Schweitzer, "Results of an initial attempt to cryptanalyze the NBS data encryption standard," Information System Lab. report SEL 76-042, Stanford University, 1976.

[34] D. Chaum and J. Evertse, "Cryptanalysis of DES with a reduced no. of rounds sequences of linear factors in block cipher," *Proc. CRYPTO - Lecture Notes in Computer Science*, vol. 218, pp. 192-211, 1986.

[35] S. Dubuc, "Linear structures of Boolean functions," *IEEE Trans. on Information Theory*, vol.16-21, p.p.440, 1998.

[36] E. Dawson and C.-k. Wu, "On the linear structure of symmetric Boolean functions," *Australasian Journal of Combinatorics*, vol. 16, pp. 239-243, 1997.

[37] P. Stanica and S. Maitra, "A constructive count of Rotation Symmetric functions," *Information Processing Letters*, vol. 88, pp. 299-304, 2003.

[38] J. Pieprzyk and C.X. Qu, "Fast hashing and rotation-symmetric functions," *Journal of Universal Computer Science*, vol 5, no 1, pp. 20-31, 1999.

[39] P. Stanica, S. Maitra, and J. Clark, "Results on rotation symmetric bent and correlation immune Boolean functions," *Proc. FSE - Lecture Notes in Computer Science*, vol. 3017, pp. 161-177, 2004.

[40] S. Kavut and M. D. Ycel, "Generalized rotation symmetric and dihedral symmetric Boolean functions - 9 variable Boolean functions with nonlinearity 242," *Proc. AAECC - Lecture Notes in Computer Science*, vol. 4851, pp. 321-329, 2007.

[41] S. Sarkar and S. Maitra, "Construction of rotation symmetric Boolean functions on odd number of variables with maximum algebraic immunity," *Proc. AAECC - Lecture Notes in Computer Science*, vol. 4851, pp. 271-280, 2007.

[42] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. on Information Theory*, vol. 15, no. 1, pp. 122-127, 1969.

[43] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. on Information Theory*, vol. 30, no. 5, pp.776-780, 1984.

[44] M. Liu, P. Lu and G. Mullen, "Correlation-immune functions over finite fields," *IEEE Trans. on Information Theory*, vol. 44, no. 3 pp. 1273-1276, 1998.

[45] Y. Hu and G. Xiao, "Resilient functions over finite fields," *IEEE Trans. on Information Theory*, vol. 49, no. 8, pp. 2040-2046, 2003.

[46] Y. Li and T. Cusick, "Strict avalanche criterion over finite fields," *Mathematical Cryptology*, vol. 1, pp. 65-78, 2005.

[47] P. Kumar, R. Scholtz and L. Welch, "Generalized bent functions and their properties," *Combinatorial Theory*, Ser. A, vol. 40, no. 1, pp. 90-107, 1985.

[48] A. Youssef, "Generalized hyper-bent functions over $GF(p)$," *Discrete Applied Mathematics*, vol. 155, no. 8, pp. 1066-1070, 2007.

[49] K. Khoo, G. Gong and D. Stinson, "A New characterization of semi-bent and bent functions on finite fields," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 279-295, 2006.

[50] Y. Li, "Results on rotation symmetric polynomials over $GF(p)$," *Information Sciences*, vol. 178, no. 1, pp. 280-286, 2008.

[51] T. Cusick, Y. Li and P. Stanica, "Balanced symmetric functions over $GF(p)$," *IEEE Trans. on Information Theory*, vol. 54, no. 3, pp. 1304-1307, 2008.

[52] A. Canteaut, C. Carlet, P. Charpin and C. Fontaine, "Propagation characteristics and correlation-mmunity of highly nonlinear Boolean functions," *Proc. EUROCRYPT - Lecture Notes in Computer Science*, vol. 1807, pp. 507-522, 2000.

[53] C. Wu and V. Varadharajan, "Public key cryptosystems based on Boolean permutations and their applications," *International Journal of Computer Mathematics*, vol.74, no. 2, pp. 167-184, 2000.

[54] B. Kaliski, "Considerations for new public-key algorithms," *Network Security*, vol. 2000, no. 9, pp. 9-10, 2000.

[55] H. Imai and T. Matsumoto, "Algebraic methods for constructing asymmetric cryptosystems," *Proc. AAECC, J. Calmet, Ed. - Lecture Notes in Computer Science*, vol. 229, pp. 108-119, 1985.

[56] J. Patarin and L. Goubin, "Trapdoor one-way permutations and multivariate polynomials," *Proc. ICISC - Lecture Notes In Computer Science*, pp. 356-368, 1997.

[57] G. Unnikrishnan, J. Joseph, and K. Singh, "Optical encryption by double-random phase encoding in the fractional Fourier domain," *Optics Letters*, vol. 25, pp. 887-889, 2000.

[58] R. Tao, J. Lang and Y. Wang, "Optical image encryption based on the multiple-parameter fractional Fourier transform," *Optics Letters*, vol. 33, pp. 581-583, 2008.

[59] C. Shih, "Fractionalization of Fourier transform," *Optics Communications*. vol. 118, no. 5-6, pp. 495-498, 1995.

[60] Z. Liu, H. Zhao and S. Liu, "A discrete fractional random transform ," *Optics Communications*, vol. 255, pp. 357-365, 2005.

[61] J.Vilardy, J.Calderon, C. Torres and L. Mattos, "Digital images phase encryption using fractional Fourier transform," *Proc. Electronics, Robotics and Automotive Mechanics*, pp.15-18, 2006.

[62] Y. Zhang and F. Zhao, "The algorithm of fractional Fourier transform and application in digital image encryption," *Proc. Information Engineering and Computer Science ICIECS*, pp. 1-4, 2009.

[63] H. Yoshimura, and R. Iwai, "New encryption method of 2D image by use of the fractional Fourier transform," *Proc. Signal Processing ICSP*, pp. 2182-2184, 2008.

[64] S. Pei and W. Hsue, "The multiple-parameter discrete fractional Fourier transform," *IEEE Signal Process. Letters*, vol. 13, no. 6, pp. 329-332, 2006.

[65] N. Du, S. Devineni and A.M. Grigoryan, "Mixed Fourier transforms and image encryption," *Proc. IEEE Systems, Man, and Cybernetics*, pp. 547-552, 2009.

[66] Y. Zhou, K. Panetta and S. Agaian, "Image encryption using discrete parametric cosine transform," *Proc. Signals, Systems and Computers, Record of the Forty-Third Asilomar*, pp. 395-399, 2009.

[67] X. Meng, R. Tao and Y. Wang, "The fractional Fourier domain analysis of decimation and interpolation," *Science in China, Ser.F*, vol. 50, pp. 521-538, 2007.

[68] X. Meng, R. Tao and Y. Wan, "Fractional Fourier domain analysis of cyclic multirate signal processing," *Science in China, Ser E*, vol. 51, pp. 803-819, 2008.

[69] B. Schneier, *Applied Cryptography*, 2nd ed. New York,Wiley, 1996.

[70] Y. Xiao, H, Zhang, Q. Ran, J. Zhang and L. Tan, "Image encryption and two dimensional discrete M-parameter fractional Fourier transform," *Proc. Congress on Image and Signal Processing CISP*, pp. 1-4, 2009.

[71] H. Cheng and L. Xiaobo, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2439-2451, 2000.

[72] N. Bourbakis and A. Dollas, "Scan-based compression-encryption hiding for video on demand," *IEEE Multimedia Mag.*, vol. 10, pp. 79-87, 2003.

[73] L. Marvel, C. Boncelet and C. Retter, "Spread spectrum image steganography," *IEEE Trans. Image Process.*, vol. 8, no. 8, pp. 1075-1083, 1999.

[74] F. Petitcolas, R. Anderson and M. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062-1078, 1999.

[75] G. Blakley, "Safeguarding cryptography keys," *Proc. AFIPS National Computer Conference*, vol. 48, pp. 313-317, 1979.

[76] C. Chang and R. Hwang, "Sharing secret images using shadow codebooks," *Information Sciences*, vol. 111, pp. 335-345, 1998.

[77] R. Wang and C. Su, "Secret image sharing with smaller shadow images," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 551-555, 2006.

[78] C. Thien and J. Lin, "Secret image sharing," *Computers and Graphics*, vol. 26, no. 1, pp. 765-770, 2002.

[79] Y. Wu, L. Thien and J. Lin, "Sharing and hiding secret images with size constrain," *Pattern Recognition*, vol. 37, pp. 1377-1385, 2004.

[80] J. Feng, H. Wu, C. Tsai and Y. Chu, "A new multi-secret image sharing scheme using Lagranges interpolation," *Systems and Software*, vol. 76, pp. 327-329, 2005.

[81] A. De Santis and B. Masucci, "New results on non-perfect sharing of multiple secrets," *Systems and Software*, vol. 80, pp. 216-223, 2007.

[82] M. Rey, "A matrix-based secret sharing scheme for images," *Proc. CIARP - Lecture Notes In Computer Science*, vol. 5197, pp. 635-642, 2008.

[83] G. Alvarez, L. Hernández and A. Martín, "A new secret sharing scheme for images based on additive 2-dimensional cellular automata," *Proc. IbPRIA - Lecture Notes in Computer Science*, vol. 3522, pp. 411-418, 2005.

[84] G. Alvarez, A. Hernández Encinas, L. Hernández Encinas and A. Martín del Rey, "A secure scheme to share secret color images," *Computer Physics Communications*, vol. 173, pp. 9-16, 2005.

[85] J. Storer, *Data Compression: Methods and Theory,* Computer Science Press, Rockville, MD, 1988.

[86] ZLIB Specification, http://www.zlib.org.

[87] G. Bard, "Algorithms for the solution of linear and polynomial systems of equations over finite fields, with applications to cryptanalysis," PhD thesis, University of Maryland at College Park, USA, 2007.

[88] C. Studholme, I. Blake, "Random matrices and codes for the erasure channel," *Algorithmica*, vol. 56, no. 4, pp. 605-620, 2010.

[89] S. William, *Sage Mathematics Software (Version 4.4.2)*. The Sage Group, 2010. http://www.sagemath.org.

[90] E. Elsheh, A. Ben Hamza and A. Youssef, "On the nonlinearity profile of cryptographic Boolean functions," *Proc. IEEE CCECE*, pp. 1767-1770, 2008.

[91] E. Elsheh and A. Ben Hamza, "A group key agreement protocol using bilinear pairing," *Proc. IEEE CCECE*, pp. 561-564, 2008.

[92] E. Elsheh, "On the linear structures of cryptographic rotation symmetric Boolean functions," *Proc. IEEE Symposium on Trusted Computing*, pp. 2085-2089, 2008.

[93] E. Elsheh and A. Youssef, "Cryptanalysis of a public key cryptosystem based on Boolean permutations," *Journal of Discrete Mathematical Sciences and Cryptography*, submitted, 2010.

[94] E. Elsheh and A. Ben Hamza, "Comments on matrix-based secret sharing scheme for images," *Proc. CIARP - Lecture Notes in Computer Science*, vol. 6419, pp. 169-175, 2010.

[95] E. Elsheh and A. Youssef, "On the security of image encryption schemes based on parameterized transformations," *Proc. IEEE Symposium on Signal Processing and Information Technology*, 2010.

[96] E. Elsheh and A. Ben Hamza, "Secret sharing of 3D models using Blakely scheme," *Proc. 25th Biennial Symposium on Communications QBSC*, pp. 92-95, 2010.

[97] E. Elsheh and A. Ben Hamza, "Robust approaches to 3D object secret sharing," *Proc. ICIAR - Lecture Notes in Computer Science*, vol. 6111, pp. 326-335, 2010.

[98] E. Elsheh and A. Ben Hamza, "Secret sharing approaches for 3D object encryption," *Journal of Expert Systems with Applications*, accepted, 2011.