NONLINEARITY PROPERTIES OF THE MIXING OPERATIONS USED IN THE BLOCK CIPHER IDEA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAMDİ MURAT YILDIRIM

93046

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MATHEMATICS

SEPTEMBER 2000

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Tayfur Öztürk
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Okay Çelebi
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ersan Akyıldız
Supervisor

Examining Committee Members

Prof. Dr. Ersan Akyıldız

Assoc. Prof. Dr. Melek D. Yücel

Assoc. Prof. Dr. Gülin Ercan

Assoc. Prof. Dr. Ali Doğanaksoy

Asists. Prof. Dr. Ferruh Özbudak

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ABSTRACT

NONLINEARITY PROPERTIES OF THE MIXING OPERATIONS USED IN THE BLOCK CIPHER IDEA

Yıldırım, Hamdi Murat

M.S., Department of Mathematics

Supervisor: Prof. Dr. Ersan Akyıldız

September 2000, 75 pages

The twisted modular multiplication and addition are two of the mixing operations of the block cipher IDEA. These are used to define so-called Multiplication-Addition (MA) structure of IDEA. Several computer programs are used to study the nonlinearity properties of these mixing operations and MA structure of IDEA in the sense of Nyberg and Matsui. Out of these calculations it is observed that the nonlinearities become zero for some key points. From this observation, we change the MA structure slightly to remove these cases. This is given us to define a new structure, which is called RMA structure. In this thesis, we have compared the nonlinearity values of MA, RMA and their compositions. In the light of these comparisons, we propose a slightly modified version of IDEA, which we call RIDEA and this seems to give us a more secure block cipher than IDEA.

Keywords: BlockCiphers, Nonlinearity, Cryptanalysis.

ÖZ

BLOK ŞİFRELEME SİSTEMİ IDEA' DA KULLANILAN KARIŞTIRMA OPERASYONLARININ DOĞRUSALSIZLIK ÖZELLİKLERİ

Yıldırım, Hamdi Murat

Yüksek Lisans, Matematik Bölümü

Tez Yöneticisi: Prof. Dr. Ersan Akyıldız

Eylül 2000, 75 sayfa

Şifreleme sistemi IDEA'nın karıştırma operasyonlarından ikisi değiştirilmiş moduler toplama ve çarpmadır. Bunlar IDEA'nın Toplama-Çarpma (MA) yapısını tanımlamada kullanılmaktadır. Birkaç bilgisayar programı kullanılarak bu karıştırma operasyonlarının ve IDEA'nın MA yapısının doğrusalsızlık özellikleri Nyberg and Matsui manasında çalışıldı. Bu hesaplamaların sonucunda bazı anahtar noktalarında doğrusalsızlıkların sıfır olduğu gözlendi. Bu gözlemden, MA yapısını bu durumları kaldırmak için biraz değiştirdik. Bu bize RMA yapısı olarak tanımlanan yeni bir yapı verdi. Bu tezde MA, RMA yapılarını ve onların bileşkelerinin doğrusalsızlıklarını karşılaştırdık. Bu karşılaştırmaların ışığı altında, RIDEA olarak adlandırdığımız IDEA'nın biraz değişikliğe uğramış bir versiyonunu öneriyoruz ve bunun bize IDEA'dan daha güvenli bir sistem verdiği görülüyor.

Anahtar Kelimeler: Blok şifreleme sistemleri, doğrusalsızlık, Kriptanaliz.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Professor Dr. Ersan Akyıldız for his motivation, guidance, encouragement and insight throught the preparation of this thesis. I also thanks him for his suggestion to study Cryptology.

I also express great appreciation to my friends who were with me in my hard times.

Finally, I express my gratitude to my family for their endless support and encouragement.

TABLE OF CONTENTS

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

CHAPTER 1
INTRODUCTION

In 1991 Lai, Massey and Murhpy [13] introduced the block cipher IDEA (International Data Encryption Algorithm) as a slightly modified version of PES (Proposed Encryption Standard)to increase immunity against differential cryptanalysis. IDEA and PES are based on the design concept of "mixing (arithmetic) operations from different algebraic groups." IDEA encrypts 64-bit plaintext blocks to 64-bit ciphertext blocks with 128-bit key blocks and used within the popular encryption program PGP (Pretty Good Privacy).

Highly nonlinear functions are important for the design of cryptographic transformations since they provide the required diffusion. Nyberg [21] introduced a measure for the nonlinearity based on the Hamming distance from the set of affine functions. This measure involves all nontrivial linear combinations of the coordinate functions of a vector function. The nonlinearity of a permutation and its inverse are exactly the same.

Matsui [18] discovered a known-plaintext attack, called Linear Cryptanalysis which analyzes iterated ciphers by finding a statistical linear relation for the nonlinear parts such as S-Boxes. Pieprzyk, Charnes and Seberry [25] explained the relation between Nyberg's nonlinearity measure and linear approximation tables for S-Boxes.

The designers of IDEA expressed the mixing operations $\odot$ and $\boxplus$ as the functions over the $\mathbb{Z}_{2^n}$ and $\mathbb{Z}^*_{2^n+1}$, respectively and considered their nonlinearity in terms of these expressions. In this thesis, we study the nonlinearity properties of these operations by using the nonlinearity measure of Nyberg. For this purpose, we view them as functions on the corresponding vector spaces $\mathbb{Z}_2^n(\cong \mathbb{Z}_{2^n})$ and $\mathbb{Z}_2^n(\cong \mathbb{Z}^*_{2^n+1})$ respectively. We prove that the nonlinearity of the vector function corresponds to the multiplication operation $\odot$ is zero for some fixed points. Moreover we observe the effects of these points on the linearity of MA (Multiplication-Addition) structure of IDEA. By changing MA structure slightly we introduce so-called RMA (Reverse MA) structure. This strengths the nonlinearity property. In fact the nonlinearities of RMA, MA, several compositions of MA and RMA structures are calculated and compared in the sense of Nyberg and Matsui. In the light of these observations we propose a block cipher RIDEA (Reverse IDEA).

The thesis is organized as follows: in the remaining part of this chapter, the basic definitions and notations are provided. Chapter 2 gives the information about block ciphers. In Chapter 3, the detailed description of PES and IDEA are given. Chapter 4 presents our study. Chapter 5 covers the conclusion.

1.1 Notation and Terminology

Cryptography is the study of mathematical techniques for sending messages in concealed form over an insecure channel (telephone line, computer networks, etc) so that any intruder cannot comprehend to these messages and intended user can read them by removing concealment of these messages. From now on, we have called sent and received messages as *plaintext* and *ciphertext* respectively. They are composed of characters that can be not only letters in one alphabet but also numerals, punctuation marks , or any other symbol commonly used. In addition we can represent each characters of plaintext p=$(p_1 p_2 \dots p_m)$ and ciphertext c=$(c_1 c_2 \dots c_m)$ by 1's and 0's. For example, let us take all characters which constitute of plaintext and ciphertext as English Alphabet's letter 'A'-'Z' and punctuation marks '.', '!','?'. For each symbol, we give a number and represent them in base 2 in a following way:

$$A \leftrightarrow 0 \leftrightarrow 00000$$

$$B \leftrightarrow 1 \leftrightarrow 00001$$

$$\vdots$$

$$Z \leftrightarrow 25 \leftrightarrow 11001$$

$$. \leftrightarrow 26 \leftrightarrow 11010$$

$$! \leftrightarrow 27 \leftrightarrow 11011$$

$$? \leftrightarrow 28 \leftrightarrow 11100$$

Encryption is the process of converting plaintext to a ciphertext. *Decryption* is the reverse process of encryption. *Encryption function* E takes any plaintext and gives us a ciphertext. In other words, it is a map from the set of P of all possible plaintext to the set C of all possible ciphertext. We

shall always assume that E is a 1-1 and onto function. This means that for each ciphertext, we can obtain one and only one plaintext. The inverse of E is the *Decryption function* $D = E^{-1}$ converts a given ciphertext to plaintext. Given a key $k \in K$ which has the same structure as plaintext and ciphertext, we obtain unique Encryption E_k and Decryption function D_k . Here we denote K is the set of all possible keys.

A system which consists of P, C, K, E_k and D_k is called *cryptosystem (cipher)*. There are two types of cryptosystem; *symmetric and asymmetric.* In a symmetric cryptosystem, For the encryption and decryption processes, the same key is used. More precisely, $D_k \circ E_k(\mathrm{p})=\mathrm{p}$ for any plaintext $p \in P$ and $k \in K$. As opposite to this cryptosystem, an asymmetric cryptosystem uses two different keys (usually one of them is publicly known and the other one is secret) for the encryption and decryption.

Graphically,

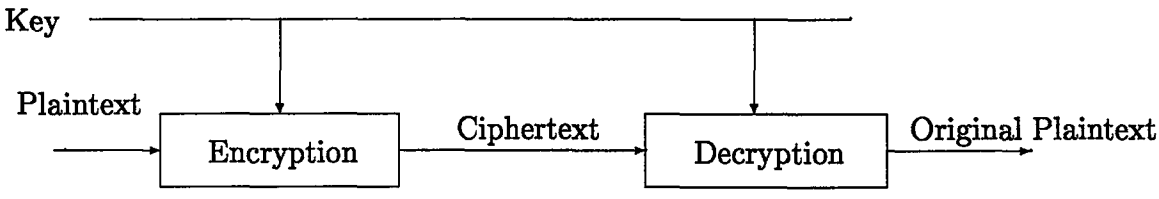


Figure 1 Symmetric Cryptosystem

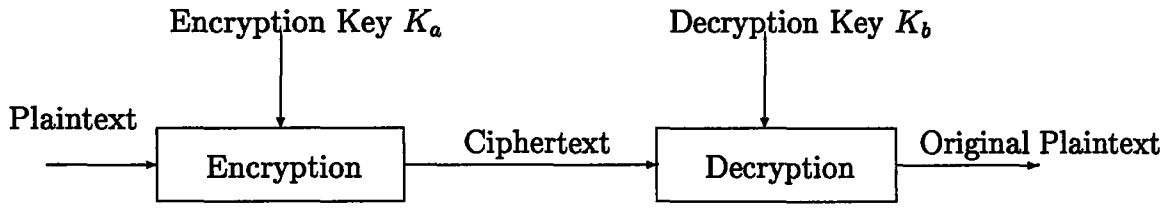


Figure 2 Asymmetric Cryptosystem

Cryptology is a field consisting the study and the research of methods for the encryption and decryption. The name cryptology comes from Greek words cruptos (hidden) and logos (study, science). Cryptology which is known as the science of concealing is the combination of two fields: Cryptography and Cryptanalysis. *Cryptanalysis* is the set of techniques for obtaining plaintext from ciphertext without knowing the secret keys. A *Cryptanalyst* analyzes the encryption process of cryptosystems and encrypted plaintexts by using cryptanalysis to find the original plaintexts. *Cryptographer* who creates cryptosystems using cryptography.

CHAPTER 2

BLOCK CIPHERS

Symmetric cryptosystem sometimes is referred as conventional cryptosystem or one-key encryption was the only type of encryption in use prior to introduction of public-key encryption in 1976 by Diffie and Hellman. There are two kinds of symmetric cryptosystem, stream ciphers and block ciphers. In stream ciphers, from a short string of key bits, pseudo-random sequence of bits is generated by a keystream generator and is added bitwise to modulo 2 to the plaintext to generate random-looking sequence of bits, namely ciphertext. By a stream cipher, from the same key used at a different time, the encryption of the fix plaintext bit does not necessarily yield the same ciphertext bit. In block ciphers, n-bit plaintext blocks encrypted to n-bit blocks using the k-bit key blocks by an invertible function where n and k is the length of blocks. By a block cipher, for each time the same ciphertext block is obtained if the same plaintext block encrypted with the same key. From now on w use the following notation. $\mathbb{Z}_2$ is the field with two element $\{0,1\}$, $\mathbb{Z}_2^n = \mathbb{Z}_2 \times \ldots \times \mathbb{Z}_2 = \{(x_{n-1}, \ldots, x_0) : \ x_i \in \mathbb{Z}_2\}$.

Definition 2.0.1 *A n-bit block cipher is a function* $E : \mathbb{Z}_2^n \times K \to \mathbb{Z}_2^n$ *such that for every* $k \in K$, $E(p,k)$ *is an invertible function of* p *from* $\mathbb{Z}_2^n$ *to* $\mathbb{Z}_2^n$. *This function is called encryption function. The inverse of the encryption function* $E(p,k)$ *is called the decryption function which is denoted by* $D(c,k)$ *(i.e.* $c = E(p,k)$*). Here* K *is an arbitrary finite set.*

Two important classes of block ciphers are substitution ciphers and transposition ciphers. Product cipher is the combination of these ciphers. In the following sections their definitions and some examples are given.

2.1 Substitution Ciphers

Substitution Ciphers are block ciphers that replace every plaintext character by some ciphertext character. There are four kinds of substitution ciphers: Simple (or mono-alphabetic) substitution, Polyalphabetic substitution, Homophonic substitution and Polygram substitution. Here we only consider substitution ciphers of first two of these kinds.

2.1.1 Simple substitution

Let $\mathbf{A}$ be an alphabet of q symbols (here $\mathbf{A}$ can be some fixed character alphabet such as $\mathbf{A} = \{A, B, \ldots, Z\}$. Suppose the ciphertext $c = (c_1 c_2 \ldots c_m)$ and plaintext $p = (p_1 p_2 \ldots p_m)$ consists of characters belonging to $\mathbf{A}$ with length m. A simple substitution encrypts given plaintext to a ciphertext using a permutation e over $\mathbf{A}$ with encryption function $E_e(p) = (e(p_1)e(p_2)\ldots e(p_m)) = (c_1 c_2 \ldots c_m)$. In other words, each symbol in a $p = (p_1 p_2 \ldots p_m)$ m-tuple, substituted it by another symbols from $\mathbf{A}$ according to some fixed permutation e. For decryption of $c = (c_1 c_2 \ldots c_m)$, inverse permutation of e, $e^{-1} = d$ is used; $D_d(c) = (d(c_1)d(c_2)\ldots d(c_m))$.

Example 2.1.1 *The earliest known use of a substitution cipher, and simplest, was by Julius Caesar. In the Caesar cipher, each letter is translated to the letter standing three places further up the alphabet. Caesar used a shift of 3, so*

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

that plaintext letter p was encrypted as ciphertext letter c by $c = E_e(p) = p+3$.

For example,

Plaintext: TODAY IS YESTERDAY

Ciphertext: wrgdb lv bhvwhsgdb

A full translation chart of Caesar cipher is

Plaintext Letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ciphertext Letters: defghijklmnopqrstuvwxyzabc

The general Caesar algorithm is $c = E_e(p) = (p + k)$ *mod* (26) *where k takes on a value in the range 1 to 25. The decryption is* $p = D_d(c) = (c - k)$ *mod* (26).

2.1.2 Polyalphabetic substitution

A polyalphabetic substitution cipher is a block cipher with block length t over an alphabet **A**, given by

$$E_e(p) = (e(p_1)e(p_2)\dots e(p_m)) = (c_1c_2\dots c_m)$$

under the key $e = (s_1, s_2, \dots, s_m)$ belongs to the key space K consisting of all ordered sets of m permutations such that each permutation is defined on **A**. The decryption key related with $e = (s_1, s_2, \dots, s_m)$ is $d = (s_1^{-1}, s_2^{-1}, \dots, s_m^{-1})$.

Example 2.1.2 *Let us choose* $e = (s_1, s_2, s_3)$ *and* $\mathbf{A} = A, B, \dots, Z$, *where* s_1 *sends each letter to three positions to its right in the alphabet,* s_2 *to the seven positions to its right, and* s_3 *ten positions to its right. If we have plaintext* $p =$ *WHA TIS THE KEY then* $c = E_e(p)$ *=ZOK WPC WOO NLI .*

2.1.3 Transposition Ciphers

Let $K = S_m$ be the set of all permutation on the set $\{1, \ldots, m\}$. For each $k \in K$, the encryption transformation $E_k(p) = (p_{k(1)}, p_{k(2)}, \ldots, p_{k(m)})$ taking m-block $p = (p_1 p_2 \ldots p_m)$ to m-block $c = (p_{k(1)}, p_{k(2)}, \ldots, p_{k(m)}$ is called a transposition cipher. The decryption for E_k is nothing but $D_k(c) = E_{k^{-1}}(c)$, where k^{-1} is the inverse of $k \in S_m$.

Example 2.1.3 *Let us consider a simple transformation cipher with m=6 and e= (5 4 1 2 6 3). The plaintext p=CAESAR is encrypted to c=ASCARE. Decryption uses the inverse permutation d= (3 4 6 2 1 5).*

2.1.4 Product Ciphers

As we have seen in the previous two sections, a substitution cipher merely replaces each plaintext letter with another letter from the same alphabet, the particular substitution rule being determined by the secret key. It makes difficult the analysis of the distribution of characters in the ciphertext to make known the information related with plaintext as well as the key. Transformation cipher reduces the redundancy in the plaintext alphabet by spreading the information from the individual plaintext or the key over the ciphertext. However, each of these ciphers has some weaknesses. For example, the cryptanalyze of a simple substitution cipher is easy since their frequency distribution reflects the distribution of the underlying plaintext alphabet. Polyalphabetic substitutions seem to be more secure that simple substitutions since it flattens the frequency distribution of the plaintext significantly, though their cryptanalyze

is feasible by using the powerful techniques such as Kasiski Method and Index of Coincidence. By these methods, the number of alphabet used in encryption is determined, the ciphertext is divided into pieces and each of them is considered as a simple substitution. The cryptanalyze of a transposition cipher is made possible by computing the letter frequencies because it keeps the number of characters of a given plaintext. Hence by the combination of substitution and transposition ciphers, it is possible to make strong ciphers. These ciphers are called as *product ciphers*, which are the combination of n transformations $E_{k_1}E_{k_2}\dots E_{k_t}$ where $t \geq 2$ and each $E_{k_i}, i \in \{1,\dots,t\}$ is either a substitution or a transposition cipher. *A round* is the composition of a substitution and transposition cipher. A product cipher is called an *iterated cipher* if the plaintext is encrypted by iteratively applying a round function several times. In each round, the same transformation is applied to input using a subkey derived from the provided secret key by a key schedule algorithm. One of the well-known class of iterated block ciphers is *Fiestel Ciphers*:

Definition 2.1.4 *A Feistel Cipher is an iterated block cipher with block size 2n, which maps two n-bit block L_0 and R_0 to a pair (R_r, L_r), namely ciphertext by r-rounds and i^{th} round is*

$$g_{K_i}^{(i)} : (L_{i-1}, R_{i-1}) \rightarrow (L_i, R_i)$$

where $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$, F is any transformation mapping two given n-bit block (X_0, Y_0) to a n-bit block Z_0 and K_i is derived from the secret key by using a key scheduling algorithm.

In a Fiestel cipher, the function F may be a product cipher and it does not need to be invertible for the decryption transformation, which is same as the encryption transformation. The ciphertext is taken as input and subkeys K_i used in reverse order during decryption. Hence both encryption and decryption transformation of a Fiestel cipher use only one algortihm. In many symmetric block ciphers in use, the structure of a Fiestel cipher is used. The most well known example of such ciphers is Data Encryption Standard (DES). It is also possible to design iterative ciphers that are not Fiestel ciphers such that the encryption and decryption transformations are structurally the same. One of the popular example is International Data Encryption Algorithm (IDEA), which we will describe in next chapter.

2.2 Cryptanalysis of Block Ciphers

For the evaluation of the security of block ciphers, there are two general assumptions:

- The cryptanalyst (the opponent) has access all data transmitted over the ciphertext channel.
- He knows all details of the cryptosystem being used except the secret key. This is usually referred as to *Kerckhoff's principle*

Cryptanalytic attacks are often classified according to differing assumptions about resources are avaible to the cryptanalyst. The most important classes of atttacks for symmetric cryptosystems are:

1. *Ciphertext-only attack* The cryptanalyst has only know the ciphertext. For this type of attack, it is not easy to device a successful attack for a

secure cipher.

2. *Known-plaintext attack* In this attack, the assumption is that the cryptanalyst knows the plaintext $P_1, P_2, \ldots, P_m$ and the corresponding $E_K(P_1), E_K(P_2), \ldots, E_K(P_m)$ ciphertexts.

3. *Chosen-plaintext attack* The cryptanalyst have opportunity to choose a number of plaintexts to be encrypted and see the corresponding ciphertexts.

4. *Chosen-ciphertext attack* The cryptanalyst can choose set of ciphertexts for decryption and receive the corresponding plaintexts.

In the following sections, we shall describe some of the cryptanalytic attacks that are applicable to a wide range of block ciphers.

2.2.1 Exhaustive Key Search

For an n-bit block cipher with k-bit key, exhaustive key search is the process of exhaustively testing each possible candidates for the key k until the correct one is found. The resistance for exhaustive key search is an important measure of security of a block cipher,but it is not only requirement for security. In this case of a known-plaintext attack, the whole key space is progressed, a fixed ciphertext C is decrypted each trial key and those keys which do not give the known plaintext P is eliminated. Exhaustive search can be also applicable to a ciphertext-only attack when the underlying plaintext is known to contain redundancy.

2.2.2 Differential Cryptanalysis

Differential Cryptanalysis is the very well-known attack on iterative block ciphers, initially introduced by Murphy [24] in an attack on FEAL-4, but in 1990 this method (original version) was improved and published by Eli Biham and Adi Shamir. Differential Cryptanalysis is a chosen plaintext attack and analyzes the effect of the difference of a pair of plaintexts on the difference of ciphertext pairs which are the outputs of rounds in an iterative cipher to reveal information about the secret key. In 1992, Eli Biham and Adi Shamir [3] represented the improved version of differential cryptanalysis to attack the full 16-round DES in 2^{37} and tiny space by analyzing 2^{36} ciphertexts obtained from 2^{47} chosen plaintexts. This is the first attack on DES with less work force than exhaustive key search requires. Differential Cryptanalysis is also successfully applied to analyze more recent block ciphers such as FEAL, Khafre, REDOC-II, LOKI and Lucifer. According to Don Coppersmith who was one of members of team working on Lucifer and DES, methods of Differential Cryptanalysis was known by that team in 1974 while designing DES. Hence S-Boxes and permutations were prepared to defeat such attacks [7].The IBM team had to conceal this information as a secret for 18 years for national security reasons.

An iterated n-round cipher consists of a cryptographically weak round function which takes previous round function's ouputs and a subkey(s) calculated via a key scheduling algorithm as input, iterates n times. Differential cryptanalysis is mainly focus on that function since for a given plaintext under the secret key, random ciphertext is obtained but for a pair of plaintexts with

a particular difference, the difference of the corresponding ciphertext pairs can be observable by non-neglible probability. We define "difference" ΔX between plaintexts (ciphertexts) pair, X and X' as

$$\Delta X = X \otimes (X')^{-1}$$

where $\otimes$ is the group operation on the set of plaintexts (ciphertexts) and $(X')^{-1}$ is the inverse element of X' in that group. This difference does not contain the key value. For DES-like cryptosystems, the difference is taken as a fixed exclusive-or (bitwise addition over modulo 2) of the two plaintexts (ciphertexts). For others, it may change according to their structure. To analyze differential behaviour of a round function, the difference distribution table can be constructed according to difference of every plaintext and ciphertext pairs. For the extension of the single round analysis to other rounds, the notion of characteristic was introduced by Biham and Shamir [2]. An n-round characteristic is a tuple $(\alpha_0, \alpha_1, \ldots, \alpha_r)$ containing series of differences, where $\alpha_0 = \Delta P$ is the chosen difference of a pair of plaintext P_0 and P_0' which are the inputs of the first round, α_i is the difference of a pair of ciphertext that are the outputs of i^{th} round related to plaintext P_0 and P_0'. The probability of a i-round characteristic is the following conditional probability:

$$P(\Delta C_i = \alpha_i, \Delta C_{i-1} = \alpha_{i-1}, \ldots, \Delta C_1 = \alpha_1 | \Delta P = \alpha_0)$$

Here, a r-round characteristic can be either the combination of one or more round characteristic or iteration of some fixed characteristic (iterative characteristic). [4] provides the detailed information about the characteristics. In Differential Cryptanalysis, when such probabilities are computed, it is as-

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

sumed that all subkeys are independent and uniformly random to simplify the mathematical analysis. A plaintext P, P' with difference $\Delta\ P$ is a *right pair* with respect to an n-round characteristic and an independent key K if they satisfies the differences in characteristic when they are encrypted. Every pair which is not a right pair with respect to the characteristic and the independent key is called a *wrong pair.* The first step of the chosen plaintext attack, differential cryptanalysis on an n-round is to find the subkey of the last round by determining n-1 round characteristic ($\Delta\ P = \alpha_0, \Delta\ C_1 = \alpha_1, \ldots, \Delta\ C_{n-1} = \alpha_{n-1}$) which holds $\Delta\ C_{n-1}$ completely or partially with high, or nearly high probability. For every right pair P and P' with difference α_0, the occurence of candidate subkey for the last round key is counted if ciphertext pair C_{r-1} and C'_{r-1} with difference α_{r-1} is obtained from the last round ciphertext C_r and C'_r by that key. For sufficiently many chosen plaintext pairs, the above steps are repeated. Finally, the most appeared subkey(s) is taken as the actual subkey of the last round.

The notion of *differential* was introduced by Lai , Massey and Murhpy [13]. Here is its definition :

Definition 2.2.1 [13]*: An i-round differential is a couple* (α, β)*, where* α *is the difference of a pair distinct palintext* P *and* P' *and where* β *is a possible difference for the resulting* i^{th} *round outputs* C_i *and* C'_i*. The probability of an i-round differential* (α, β) *is the conditional probability that* β *is the difference* $\Delta\ C_i$ *of the ciphertext pair after i rounds given that the plaintext pair* (P, P')

has difference $\Delta P = \alpha$ *when the plaintext* P *and the subkeys* $K_1, \ldots, K_i$ *are independent and uniformly random. We denote this differential probability by* $P(\Delta C_i = \beta | \Delta P = \alpha)$.

They preferred to use differentials instead of characteristic in Differential cryptanalysis for a n-round cipher. Their reason is that the indeterminate differences is not important while one is trying to the last round key of n-round cipher in differential cryptanalysis since only n-1 round difference is adequate to find that key. It is note that the probabilities of differentials will be greater than characteristics. For this reason, it is used to derive a lower bound on the complexity of a differential cryptanalysis in [13]. The probability of an i-round differential with input difference α and output difference β is the sum of probabilities of all i-round characteristics with the corresponding input and output difference.

Lai [15] presented a definition of higher order derivatives of discrete cryptographic functions that is analogous to the definition of differentiation in calculus and introduced the concept of higher order differentials . Knudsen [10] used higher order differentials to cryptanalyze ciphers probably secure aganist a differential attack using first order differentials, drew conclusions related this and showed the existence of the ciphers which are probably secure aganist a differential attack can be vulnerable to attacks using second order differentials. In [11], an extension of attacks based on higher order differentials used in [10] was given and applied to two ciphers.

Knudsen introduced the notion truncated differentials (partial differentials). For an $2n$ bit Feistel cipher, Unlike i-round differentials with difference (α, β), i-round truncated differentials predicts only subsequence of α and β, namely α' and β'. Truncated differentials are used to analyze on DES with 6 rounds with complexity of about 46 chosen plaintexts and a running time about time of 3500 encryptions. It is noted that this type of attacks seem to be useful for ciphers that have a relatively small number of rounds.

2.2.3 Linear Cryptanalysis

Linear Cryptanalysis is one of the most recent method of analysing iterated ciphers. It is essentially a known-plaintext (statistical) attack and was initially used to attack the FEAL cipher by Matsui and Yamagishi [17]. The refinement version of linear cryptanalysis was used to break 16 round DES cipher with 2^{47} known-plaintexts [18]. Matsui's paper [19] which represents the improved version of linear cryptanalysis and its application to first experimental cryptanalysis for breaking the full 16-round DES was appeared in 1994. This experiment was carried out by using twelve HP9735/ PA-RISK 99 MHZ workstations and finally the 56 secret key bits were recovered with 2^{43} known-plaintext/ciphertext pairs in fifty days.

The aim of Linear Cryptanalysis is to investigate statistical linear relations between bits of plaintexts, the ciphertexts, the secret keys to get a linear approximate expression for an entire cipher under consideration. Similar to Differential Cryptanalysis, Linear Cryptanalysis deals with nonlinear parts of

the one round of the cipher to be successfull in its analyze. To find a statistical linear relation for the nonlinear part (function), two subset A and B containing some index numbers of inputs and output bits of that part, respectively is constructed and for each input, exclusive-or of inputs bits and corresponding output bits is calculated according to each possible subset A and B. For half of the inputs, if exclusive-or values are equal to zero, then a nonlinear approximation can be obtained by the corresponding subsets A and B. For a smaller or a larger part of the inputs, if they are equal to zero, then a linear approximation can be obtained. In this way, the linear approximations having best probability p (i.e. $|p - 1/2|$ is maximal) are found for the nonlinear parts of one round of the cipher. Then for each linear approximation of these kind is extended to the round function as a linear equation. Finally linear equations for the round functions are combined to construct a linear expression for the entire cipher. The probability of that expression is calculated from the probabilities of linear equations of the round functions using the above lemma:

Lemma 2.2.2 ([18]) *(Piling-up Lemma) Let $X_i (1 \leq i \leq n)$ be independent random variables whose values are 0 with probability p_i or 1 with probability $1-p_i$. Then the probability that $X_1 \oplus X_2 \oplus \ldots \oplus X_n = 0$ is*

$$1/2 + 2^{n-1} \prod_{i=1}^{n} (p_i - 1/2).$$

Let us denote A is n-bit plaintext (ciphertext or key), $A[i]$ is the the i^{th} bit of A and $A[i, j, \ldots, k] := A[i] \oplus A[j] \oplus, \ldots, A[k]$.

For a given m-bit cipher, the linear expression is of the form

$$P[i_1, i_2, \ldots, i_a] \oplus C[j_1, j_2, \ldots, j_b] = K[k_1, k_2, \ldots, k_c]$$

where P is a randomly given plaintext, C is the corresponding ciphertext, K is the key used to encrypt P, $i_1, i_2, \ldots, i_a, j_1, j_2, \ldots, j_b$ and $k_1, k_2, \ldots, k_c$ denotes fixed bit locations. This relation holds with the probability p such that $|p - 1/2|$ is maximal and $p \neq 1/2$. Here the value of $|p - 1/2|$ gives us a measurement for effectiveness of the above expression . One key bit of $K[k_1, k_2, \ldots, k_c]$ is determined using the maximum likelihood method described in [18] if sufficient amount of plaintexts is avaible. The success rate of this methods depends on the number of plaintexts N and $|p - 1/2|$.

Matsui [18] succeeded in finding the linear expression,which is described in above, for the block cipher DES by considering the nonlinear part of its round function,namely S-Boxes.

Definition 2.2.3 *A $(m \times n)$ S-Box is a collection of n Boolean function $F_i(x)$: $\mathbb{Z}_2^n \to \mathbb{Z}_2, i = 1, \ldots, m$ in variables $x = (x_{m-1}, x_{m-2}, \ldots, x_0)$ for which*

$$S(x) = (F_1(x), \ldots, F_n(x))$$

Matsui [18] studied the linear approximation of S-Boxes and for this issue he gave the next definition:

Definition 2.2.4 ([18]) *A linear approximation table LAT for a S-Box $S(x)$ is*

$$LAT_S(\alpha, \beta) = \#\{\, x \mid 0 \leq x \leq 2^n - 1; (\bigoplus_{i=1}^{n}(x[i] \bullet \alpha[i])) = (\bigoplus_{j=1}^{m}(S(x)[j] \bullet \beta[j]))\}$$

where $\bullet$ is the bitwise AND operation, $x = \sum_{i=1}^{n} 2^{i-1} \cdot x[i]$ and $\alpha = \sum_{i=1}^{n} 2^{i-1} \cdot \alpha[i]$ $(x[i], \alpha[i] \in \mathbb{Z}_2)$

Some theorical, practical enhancement and extensions of the linear cryptanalysis have appeared since its emergence. Kaliski and Robshaw [9] presented

a extension to the linear cryptanalytic attack using multiple linear approximations. It leads to reduction in the amount of data required for a successful linear crytpanalysis of a block cipher. Knudsen and Robshaw [26] suggested an algorithm using non-linear approximations. They replaced the linear approximations used in linear cryptanalysis with non-linear approximations. Shimoyama and Kaneko [28] derived 7 quadratic relations of S-boxes of DES using Groebner Basis techniques by considering S-Boxes as Boolean polynomials. By using one of these relations, they constructed an improved algorithm for attacking 16 round DES that is a combination of the multiple linear approximation and non-linear approximation methods mentioned above. This algoritm reduces the number of texts used in Matsui's attack [19].

2.2.4 Differential - Linear Cryptanalysis

Differential - Linear Cryptanalysis is an attack which combines techniques used in linear and differential cryptanalysis. This analysis was introduced by Langford and Hellman [16]. Although linear and differential cryptanalysis are powerful attacks, they require a large amount of text to be successful at their analyzes. The aim of Langford and Hellman was to reduce this amount required in such attacks. They attacked on a reduced 8 round version of DES which requires fewer chosen-plaintexts than differential and linear cryptanalysis require. However, they recover more bits of key and they are extendible more efficiently to 16 round DES than this attack.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ABSTRACT

NONLINEARITY PROPERTIES OF THE MIXING OPERATIONS USED IN THE BLOCK CIPHER IDEA

Yıldırım, Hamdi Murat
M.S., Department of Mathematics
Supervisor: Prof. Dr. Ersan Akyıldız
September 2000, 75 pages

The twisted modular multiplication and addition are two of the mixing operations of the block cipher IDEA. These are used to define so-called Multiplication-Addition (MA) structure of IDEA. Several computer programs are used to study the nonlinearity properties of these mixing operations and MA structure of IDEA in the sense of Nyberg and Matsui. Out of these calculations it is observed that the nonlinearities become zero for some key points. From this observation, we change the MA structure slightly to remove these cases. This is given us to define a new structure, which is called RMA structure. In this thesis, we have compared the nonlinearity values of MA, RMA and their compositions. In the light of these comparisons, we propose a slightly modified version of IDEA, which we call RIDEA and this seems to give us a more secure block cipher than IDEA.

Key words: Block Ciphers, Nonlinearity, Cryptanalysis.

ÖZ

BLOK ŞİFRELEME SİSTEMİ IDEA' DA KULLANILAN KARIŞTIRMA OPERASYONLARININ DOĞRUSALSIZLIK ÖZELLİKLERİ

Yıldırım, Hamdi Murat

Yüksek Lisans, Matematik Bölümü
Tez Yöneticisi: Prof. Dr. Prof. Dr. Ersan Akyıldız
Eylül 2000, 75 sayfa

Şifreleme sistemi IDEA'nın karıştırma operasyonlarından ikisi değiştirilmiş moduler toplama ve çarpmadır. Bunlar IDEA'nın Toplama-Çarpma (MA) yapısını tanımlamada kullanılmaktadır. Birkaç bilgisayar programı kullanılarak bu kartıştırma operasyonlarının ve IDEA'nın MA yapısının doğrusalsızlık özellikleri Nyberg and Matsui manasında çalışıldı. Bu hesaplamaların sonucunda bazı anahtar noktalarında doğrusalsızlıkların sıfır olduğu gözlendi. Bu gözlemden, MA yapısını bu durumları kaldırmak için biraz değiştirdik. Bu bize RMA yapısı olarak tanımlanan yeni bir yapı verdi. Bu tezde MA, RMA yapılarını ve onların bileşkelerinin doğrusalsızlıklarını karşılaştırdık. Bu karşılaştırmaların ışığı altında, RIDEA olarak adlandırdığımız IDEA'nın biraz değişikliğe uğramış bir versiyonunu öneriyoruz ve bunun bize IDEA'dan daha güvenli bir sistem verdiği görülüyor.

Anahtar Kelimeler: Blok şifreleme sistemleri, doğrusalsızlık, Kriptanaliz.

2.2.5 The Interpolation Attack

Jakobsen and Knudsen [11] introduced an attack on block ciphers, namely the interpolation attack used to analyze block ciphers having S-Boxes that are simple algebraic functions. This method uses the Langrange interpolation formula for its analyze to construct polynomials using pairs of plaintexts and ciphertexts to find an algorithm equivalent to encryption (or decryption) with key K or recover the the last round key K of an iterated block cipher.

2.2.6 Weak Keys

For a block cipher, if the encryption E_k is identical to decryption D_k for some key value $k \in K$, i.e., $E_k^2 = E_k \circ E_k(p) = p$ for every $p \in P$. We call such keys *weak keys*. The key values k_1 and k_2 are called semi-weak keys when $E_{k_1} \circ E_{k_2}(p) = p$ for every $p \in P$, i.e. $E_{k_1} = D_{k_2}$. If the number of weak keys constitutes a large part of all possible keys, the probability of selecting one of such keys at random increased remarkably. They cause the threat to security of the block cipher when they used for encryption. On the contrary, the existence of a small set of weak keys has no influence on the security of the block cipher.

2.2.7 Algebraic Attack

Algebraic Attack is a method of cryptanalytic attack used against block ciphers that exhibit a significant amount of mathematical structure. If for each pair of key (k_1, k_2), there exists a key k_3 such that $E_{k_1} \circ E_{k_2}(p) = E_{k_3}(p)$ for

every plaintext p, then the set of all encryption transformation form a group. A block cipher having such property is considered weaker, because the use of multiple encryption will give the same level security as single encryption.

2.3 Design criteria and principles

In this section, we discuss several principles and criteria for the design of secure block ciphers.

2.3.1 Confusion and Diffusion

In 1949 Shannon suggested the principles of confusion and diffusion which are the desired and generally accepted attributes for practical ciphers. Shannon described these principles as follows:

"The method of confusion is to make the relation between the simple statistics of the ciphertext and the simple description of the key a very complex and involved one." [27]

"In the method of diffusion the statistical structure of the plaintext which leads to its redundancy is dissipated into long range statistics."[27]

Confusion is intended to make the relationship between the key and ciphertext as complex as possible to prevent attempts to recover the key.

By diffusion, the influence of a single plaintext digit is outstretched so as to hide the statistical structure of the plaintext.

As we have mentioned, two simple ciphers, transposition and substitution

ciphers are quite weak on their own. However, the usage of their combination (i.e., product cipher) is a very common approach to achieving good diffusion and confusion,which is an important issue in block cipher design.

2.3.2 Completeness, Avalanche and The Strict Avalanche Criterion

Definition 2.3.1 *For a given one-one correspondence $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$, f is said to be complete if, for every $i, j \in \{1, \ldots, n\}$, there exist two n-bit vectors X_1,X_2 such that X_1 and X_2 differ only in i^{th} bit and $f(X_1)$ differs from $f(X_2)$ in at least the j^{th} bit.*

" For a given transformation to exhibit the avalanche effect, an average of one half of the output bits should change whenever a single input bit is complemented. In order to determine whether a given $m \times n$ (m input bits and n output bits) function f satisfies this requirement, the 2^m plaintext vectors must be divided into 2^{m-1} pairs, X and X_i, such that X and X_i differ only in bit i. Then the 2^{m-1} exclusive-or sums $V_i = f(X)$ XOR $f(X_i)$ must be calculated (XOR is the bitwise addition over modulo 2). These exclusive-or sums will be referred to as avalanche vectors, each of which contains n bits, or avalanche variables. "If this procedure is repeated for all i such that $1 \leq i \leq n$, and one half of the avalanche variables are equal to 1 for each i, then the function f has a good avalanche effect. Of course this method can be pursued only if m is fairly small; otherwise, the number of plaintext vectors becomes too large. If that is the case then the best that can be done is to take a random sample of plaintext vectors X, and for each value of i calculate all the avalanche vectors

V_i. If approximately one half the resulting avalanche variables are equal to 1 for all values of i, then we can conclude that the function has a good avalanche effect." [29]

"The concepts of completeness and the avalanche effect can be combined to define a new property which we shall call the strict avalanche criterion. If a cryptographic function is to satisfy the strict avalanche criterion, then each output bit should change with a probability of one half whenever a single input bit is complemented. A more precise definition of the criterion is as follows. Consider X and X_i, two n-bit binary plaintext vectors, such that X and X_i differ only in bit i, $1 \leq i \leq n$. Let $V_i = f(Y)$ XOR $f(Y_i)$ where $Y = f(X)$, $Y_i = f(X_i)$ and f is the cryptographic transformation under consideration. If f is to meet the strict avalanche criterion, the probability that each bit in V_i is equal to 1 should be one half over the set of all possible plaintext vectors X and X_i. This should be true for all values of i." [29]

2.3.3 Sufficiently large key size

The key size of the most widely used block cipher DES is only 56, which is too short to achieve immunity aganist exhaustive key search. In 1993 Weiener [30] concluded that a machine could be built at the cost of $ 1 million that would find a DES key in an average time of about 3.5 hours. Today a block cipher with a key size of 80 bits seems to be not vulnerable to an exhaustive key search for next years. Moreover, one should be aware of the fact that every 18 months the performance of computers is doubled. National Institute for Standard and Technology (NIST) requires for a successor of the DES, the

AES (Advanced Encryption Standard) must work with three key sizes 128, 192 and 256 bits.

2.3.4 Nonlinearity

Highly nonlinear functions are indispensable to design of cryptographic transformations such as block ciphers, hash functions and stream ciphers. *"Confusion is reflected in nonlinearity of certain Boolean functions describing the cryptographic transformation. Nonlinearity is crucial since most linear systems are easily breakable."* [23]

A measurement for the nonlinearity of a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is defined by Nyberg as:

Definition 2.3.2 ([21]) *The nonnegative integer*

$$N(f) = min_{u \in \mathbb{Z}_2^n, v \in \mathbb{Z}_2} \#\{x \in \mathbb{Z}_2^n \mid f(x) \neq u^t \cdot x + v\}$$

is the Hamming distance of f from affine functions.

The concept of nonlinearity of arbitrarily vector function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ can be extended as follows:

Definition 2.3.3 ([21])

$$N(f) = min_{u \in \mathbb{Z}_2^n, w \in \mathbb{Z}_2^m, v \in \mathbb{Z}_2, u \neq 0 or w \neq 0} \#\{x \in \mathbb{Z}_2^n \mid w^t \cdot f(x) \neq u^t \cdot x + v\}$$

This measure has the following property:

Theorem 2.3.4 ([21]) *Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ be a permutation. Then*

$$N(f) = N(f^{-1})$$

Proof: ([21])

2.3.5 Immunity to Linear Cryptanalysis

Linear Cryptanalysis exploits the low nonlinearity of S-Boxes engaged by a block cipher. To immunize a S-box against linear cryptanalysis, it suffices that all entries of its LAT (Linear Approximation Table) should not to diverge too from 2^{n-1}. Alternatively, its nonlinearity should be high (near to 2^{n-1}) due to the relation between the nonlinearity of a S-box and its LAT. In fact they have shown the following: for a given $\beta = (\beta_1, \ldots, \beta_n)$, Let $S_\beta(x) : \mathbb{Z}_2^m \to \mathbb{Z}_2$ be the linear combination of the component functions of S-box $S = (F_1, \ldots, F_n) : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$:

$$S_\beta(x) = F_1(x) \cdot \beta_1 \oplus \ldots \oplus F_n(x) \cdot \beta_n.$$

Then we have

Lemma 2.3.5 ([25]) *For a fixed vector $\beta \in \mathbb{Z}^n$, the following inequality holds*

$$\forall \alpha \in \mathbb{Z}_2^n \; N(S_\beta(x)) \leq LAT_S(\alpha, \beta) \leq 2^n - N(S_\beta(x)).$$

Corollary 2.3.6 ([25]) *From a given $LAT_S(\alpha, \beta)$, it is possible to recover all the nonlinearities of $S_\beta(x)$ by selecting the minimal and the maximal values from the column $LAT_S(\alpha, \beta)$; $\alpha \in \mathbb{Z}_2^n$. Denote these two values by $LAT_S(\alpha_{min}, \beta)$ and $LAT_S(\alpha_{max}, \beta)$. Then the nonlinearities of $S_\beta(x)$ is $min(LAT_S(\alpha_{min}, \beta), 2^n - LAT_S(\alpha_{max}, \beta))$.*

Corollary 2.3.7 ([25]) *The nonlinearity of $S(x)$ is*

$$N(S(x)) = min_{\beta \in \mathbb{Z}_2^n} N(S_\beta(x)).$$

In [25] the nonlinearity of a permutation $S(x)$ was defined by using its linear approximation table in the following manner:

$$N(S(x)) = 2^{n-1} - \gamma.$$

where $\gamma = max_{\alpha,\beta=1,\ldots,2^n-1}|LAT_S(\alpha,\beta) - 2^{n-1}|$

2.3.6 Immunity to Differential Cryptanalysis

To immunize against differential cryptanalysis, the difference distribution tables of the S-boxes of a DES-like iterated block cipher must not contain entries with large values except for the first entry of the first row. In other words, the values of the difference distribution table of S-Boxes must be uniformly distributed. In addition to this requirement, the difference distribution table of an S-Box should also contain less nonzero entries as possible in its fist column to lessen the probability of the possible iterative characteristics.

In [20] it was shown that for DES-like iterated ciphers, it is feasible to find an upper bound on the probabilities of r-round differential by using a non-trivial 1-round differential with highest probability. If this probability is chosen to be small, the resistance against a differential cryptanalysis can be obtained.

In [13] and [14], the notion of *Markov Cipher*, which gives more information about the probabilities of differentials, was presented and the security of these ciphers aganist differential cryptanalysis by using Markov chain techniques was discussed. It is known that the block cipher DES, LOKI, FEAL and REDOC are Markov Ciphers [14]. For the immunity of Markov Ciphers aganist differential cryptanalysis, for r-round Markov the transition probability matrix of the homogeneous Markov chain $\Delta P = \Delta C_0, \Delta C_1, \ldots, \Delta C_r$ is

defined and their irreducibility and the eigenvalues are considered in [14]. In reality, the formulation of these requirements is not practical for a block cipher with large size. Due to this issue, by the help of the results of differential cryptanalysis of PES [13], the suggested more practical requirement for security is :

The transition probability matrix of a Markov cipher should be non-symmetric. [13],[14]

The idea behind this suggestion is that when transition matrix of a Markov cipher is the symmetric, for the one-round differential with high probability among the others, the concatenation of that differential with itself r-1 can result in the r-round characteristic with high probability that produce an r-round differential with high probability; however, this situation can be avoided by making the corresponding transition matrix non-symmetric.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

CHAPTER 3

BLOCK CIPHERS : PES AND IDEA

3.1 Proposed Encryption Standard

In 1990 the iterated block cipher Proposed Encryption Standard (PES) was proposed by J.Massey and X. Lai [12] as a replacement of DES. PES operates on 64 bits plaintext, ciphertext and 128 bits long key and consists of 8 iterated rounds and a final transformation. The rounds and final transformation is arranged to achieve the desired confusion and diffusion by successive usage of three incompatible group operation and its chosen special structure. The design of PES is completely based on the concept of "mixing operations from different algebraic groups such as multiplication modulo $2^{16}+1$, addition modulo 2^{16} and bitwise XOR (bitwise addition on modulo 2) on 16 bit blocks"

3.1.1 Description of the block cipher PES

PES completely prevents the use of any lookup or S-Boxes used as in DES. The 16-bit subblocks are connected via three different group operation of 2^{16} elements described below:

- bit-by-bit exclusive-OR (bitwise addition over modulo 2) of two 16-bit subblocks, denoted as $\oplus$

- addition of integers modulo 2^{16} where the 16-bit subblock is treated as the usual radix-two representation of an integer; the resulting operation is denoted as $\boxplus$
- multiplication of integers modulo $2^{16}+1$ where the 16-bit subblock is treated as the usual radix-two representation of an integer except that the all-zero subblock is treated as representing 2^{16}; the resulting operation is denoted as $\odot$

For the description of the encryption algorithm of PES, let n be positive integer such that 2^n+1 is prime. $\mathbb{Z}_{2^n} = \{0, 1, \ldots, 2^n - 1\}$ denotes the ring of integers modulo 2^n, $\mathbb{Z}^*_{2^n+1}$ denotes the multiplicative group of the non-zero elements of the field $\mathbb{Z}_{2^n+1}$ modulo 2^n+1. We will view $\mathbb{Z}_2^n$ as an additive group and $\mathbb{Z}_{2^n+1}$ as a multiplicative group for the rest of discussions. The operations discussed above can be viewed as three vector-valued functions $\tilde{f}, \tilde{g}, \tilde{h} : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ as follows:

Let us consider the map $v : \mathbb{Z}_{2^n} \to \mathbb{Z}_2^n = \mathbb{Z}_2 \times \ldots \times \mathbb{Z}_2$ (n-times $\mathbb{Z}_2$) and the direct mapping $d : \mathbb{Z}^*_{2^n+1} \to \mathbb{Z}_{2^n}$ given by
$v(x) = (x_{n-1}, \ldots, x_1, x_0)$ such that $x = \sum_{i=0}^{n-1} x_i \cdot 2^i$ and

$$d(x) = \begin{cases} 0 & if \quad x = 2^n \\ x & if \quad x \neq 2^n \end{cases} \tag{3.1}$$

are bijective with inverses d^{-1}, v^{-1} for each n; a_i and x_i are the i^{th} component of the vectors $v(x)$ and $v(a)$, respectively; f, g and h are the functions from $\mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ to $\mathbb{Z}_{2^n}$.

For the multiplication operation $\odot$,
$\tilde{g}(v(a), v(x)) = v(a) \odot v(x) = v(g(a, x))$ such that

$$g(a, x) = d(d^{-1}(a) \cdot d^{-1}(x) \bmod (2^n + 1)) \quad \forall a, x \in \mathbb{Z}_{2^n}; \tag{3.2}$$

For the addition operation $\boxplus$,
$\tilde{f}(v(a), v(x)) = v(a) \boxplus v(x) = v(f(a, x))$ such that

$$f(a, x) = a + x \bmod (2^n) \quad \forall a, x \in \mathbb{Z}_{2^n}; \tag{3.3}$$

For the exclusive-OR operation $\oplus$,
$\tilde{h}(v(a), v(x)) = v(a) \oplus v(x) = v(h(a, x))$ such that

$$h(a, x) = \sum_{i=0}^{n-1}((a_i + x_i) \bmod 2) \cdot 2^i \quad \forall a, x \in \mathbb{Z}_{2^n}; \tag{3.4}$$

The block cipher PES encrypts blocks of 64 bits plaintext to blocks of 64 bits ciphertext with 128 bit key. For the encryption, this cipher divides 64-bit plaintext block X into four 16 bit subblocks, X_1, X_2, X_3 and X_4 such that $X = (X_1, X_2, X_3, X_4) = (X_1^{(0)}, X_2^{(0)}, X_3^{(0)}, X_4^{(0)})$. They are transformed into four 16-bit ciphertext subblocks Y_1, Y_2, Y_3 and Y_4 by 8 iterated rounds and a final output transformation using 52 key subblocks with length 16 derived from a given 128-bit key block. The r^{th} round of PES uses the six key

subblocks $Z_1^{(r)}, Z_2^{(r)}, .., Z_6^{(r)}$, whereas the final output transformation uses four 16-bit key subblocks $Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)}$ where $r = 1, 2, .., 8$. The graph of the encryption of PES can be seen in Figure 3. In addition, the encryption algorithm can be formulated as:

$$PES(X) = A(P(E^{(8)}(E^{(7)}(\ldots(E^{(1)}(X_1^{(0)}, X_2^{(0)}, X_3^{(0)}, X_4^{(0)}))))), Z_\gamma) = Y.$$

where $E^{(r)} = P(H(MA(L(A(X^{(r-1)}, Z_\alpha)), Z_\beta))) = X^{(r)}$ is the r^{th} round and $X^{(r-1)} = (X_1^{(r-1)}, X_2^{(r-1)}, X_3^{(r-1)}, X_4^{(r-1)})$, $Z_\alpha = (Z_1^{(r)}, Z_2^{(r)}, Z_3^{(r)}, Z_4^{(r)})$, $Z_\beta = (Z_5^{(r)}, Z_6^{(r)})$, $Z_\gamma = (Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)})$,$P(X) = (X_3, X_4, X_1, X_2)$, $A(X, Z) = X \otimes Z = (X_1 \odot Z_1, X_2 \odot Z_2, X_3 \boxplus Z_3, X_4 \boxplus Z_4)$, $L(X) = (X_1 \oplus X_3, X_2 \oplus X_4)$, $H(X, T) = (X_1 \oplus T_2, X_2 \oplus T_1, X_3 \oplus T_2, X_4 \oplus T_1)$, and $MA(P_1, P_2, Z_1, Z_2) = (C_1, C_2)$ given by

$$\begin{aligned} C_2 &= ((K_1 \odot P_1) \boxplus P_2) \odot K_2. \\ C_1 &= (K_1 \odot P_1) \boxplus C_2. \end{aligned}$$

3.1.2 Key Schedule and Decryption Algorithm

As we have mention that for a given 128-bit key, 52 16-bit subblock keys are generated for the encryption. For the construction of these subblocks, the first step is to partition given 128-bit key into 8 pieces and assign them as the first 8 subblock keys of the 52 subblocks: $Z_1^{(1)}, Z_2^{(1)}, .., Z_6^{(1)}, Z_1^{(2)}, Z_2^{(2)}, .., Z_6^{(2)}, .., Z_1^{(8)}, Z_2^{(8)}, .., Z_6^{(8)}, Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)}$. Then the key under the consideration is cyclically shifted to the left by 25 positions. The resulting key block is again partitioned into eight subblocks that are assigned to the next eight subblock keys. This process is repeated until all 52 subblock keys are derived.

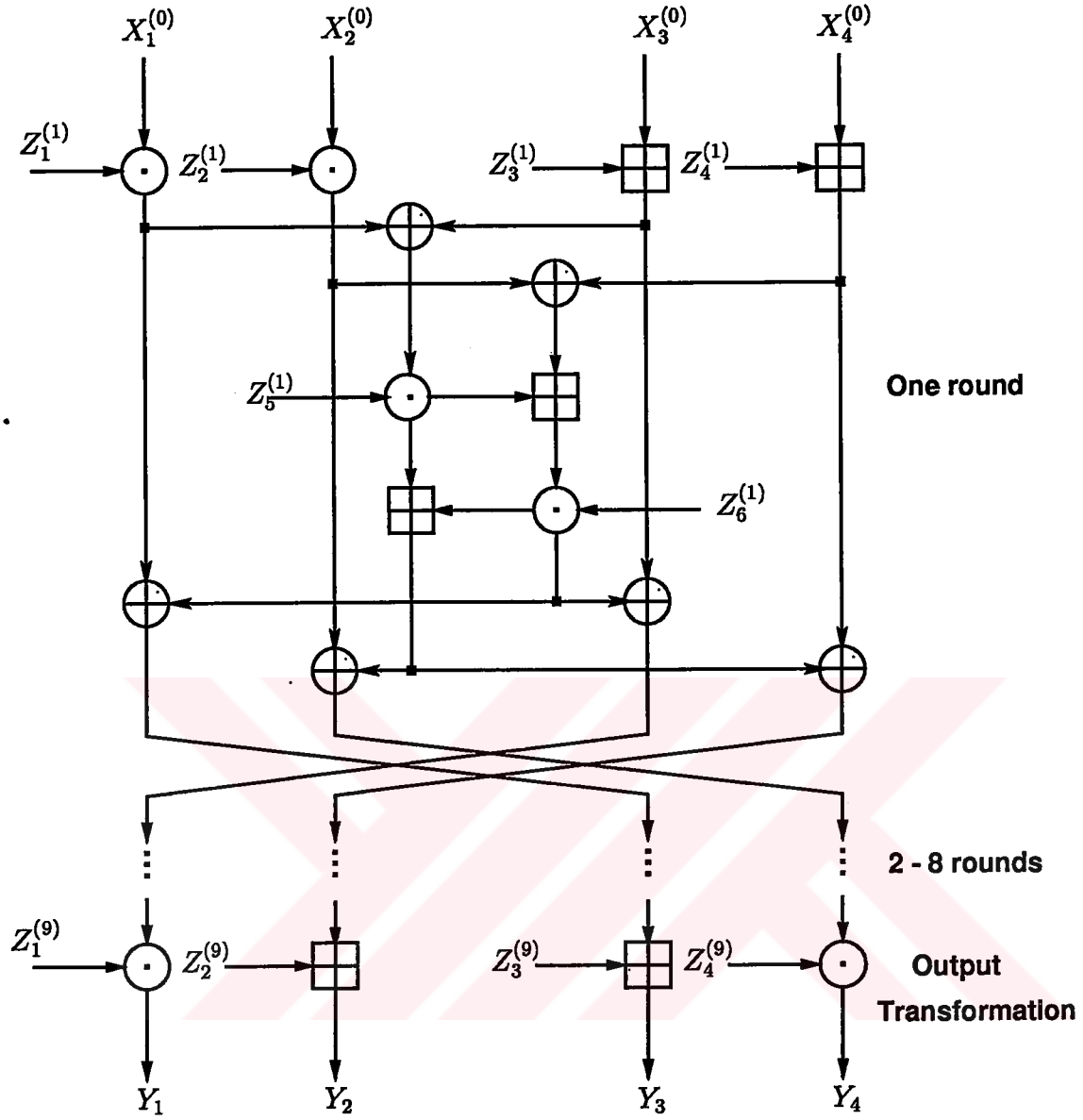


Figure 3 Computational graph for the encryption process of the PES cipher

$H(MA(L(H(MA(L(X_1, X_2, X_3, X_4), Z_1, Z_2))), Z_1, Z_2)) = (X_1, X_2, X_3, X_4)$ and $P(P(X_1, X_2, X_3, X_4)) = (X_1, X_2, X_3, X_4)$ for all Z_1 and $Z_2 \in \mathbb{Z}_2^n$. This is due to the fact that $H \circ MA \circ L$ and P is an involution transformation. For this reason, the algorithm used for the encryption of PES is also used the decryption. For more information, the interested reader can refer to [14]. The key subblocks $K_i^{(r)}$ of the decryption algorithm derived from the encryption key subblocks $Z_i^{(r)}$ are computed as:

For $r = 1, 2, 3, .., 9$,
$\left(K_1^{(r)}, K_2^{(r)}, K_3^{(r)}, K_4^{(r)}\right) = \left(Z_1^{(10-r)^{-1}}, Z_2^{(10-r)^{-1}}, -Z_3^{(10-r)}, -Z_4^{((10-r))}\right)$;
For $r = 1, 2, .., 8$,
$\left(K_5^{(r)}, K_6^{(r)}\right) = \left(Z_5^{(r)}, Z_6^{(r)}\right)$.
where Z^{-1} denotes the multiplication inverse (modulo $2^{16}+1$) of Z , i.e., $Z \odot Z^{-1} = 1$ and $-Z$ denotes the additive modulo 2^{16} of Z, i.e., $-Z \boxplus Z = 0$.

3.2 International Data Encryption Algorithm

In [13], In 1991 the differential cryptanalysis of PES was presented and with a minor change, a strengthened version of PES, IPES (Improved PES) that improves the immunity aganist differential cryptanalysis was proposed by the inventors of PES and S.Murphy. IPES was later renamed under the name IDEA (International Data Encryption Algorithm). IDEA operates on 64 bits plaintext, ciphertext and 128 bits long key and consists of 8 iterated rounds and a final transformation. As in PES, IDEA is completely based on the concept of "mixing operations from different algebraic groups".

IDEA is the one of the symmetric ciphers used in message encryption's part of PGP (Pretty Good Privacy) which is the most widely used method

encrypt and decrypt e-mail over the Internet and file storage applications. Unlike DES, exportation of IDEA is not restricted from North America since it was developed in Switzerland at ETH (Eidgenssische Technische Hochschule). Moreover no license fee is required for noncommercial use of IDEA.

3.2.1 Description of the block cipher IDEA

The block cipher IDEA encrypts blocks of 64 bits plaintext to blocks of 64 bits ciphertext with 128 bit key. As it seen the graph of the encryption of IDEA (see Figure 3 and compare with Figure 4), there is a minor changes between IDEA's and PES' encryption algorithm. In fact it is stated by the designers of IDEA that *"The only essential modification is that a different (and simpler) permutation of subblocks is used at the end of each of the first 7 rounds. The software implementation of IPES is in fact more efficient than that of PES"* [13]. Due to that change, the encryption algorithm and the formulation for the subblock keys used in PES is slightly changed in IDEA.

Similar to the block cipher PES, IDEA encrypts blocks of 64 bits plaintext to blocks of 64 bits ciphertext with 128 bit key and uses the same operations $\boxplus$, $\odot$ and $\oplus$ and the related vector functions $\tilde{f}, \tilde{g}$ and $\tilde{h}$ defined in (3.3),(3.2) and (3.4) respectively. For the encryption, this cipher divides 64-bit plaintext block X into four 16 bit subblocks, X_1, X_2, X_3 and X_4 such that $X = (X_1, X_2, X_3, X_4) = (X_1^{(0)}, X_2^{(0)}, X_3^{(0)}, X_4^{(0)})$. They are transformed into four 16-bit ciphertext subblocks Y_1, Y_2, Y_3 and Y_4 by 8 computationally identical rounds and a final output transformation using 52 key subblocks with length 16 derived from a given 128-bit key block. The

r^{th} round of IDEA's input and output are $(X_1^{(r-1)}, X_2^{(r-1)}, X_3^{(r-1)}, X_4^{(r-1)})$ and $(X_1^{(r)}, X_2^{(r)}, X_3^{(r)}, X_4^{(r)})$. It uses the six key subblocks $Z_1^{(r)}, Z_2^{(r)}, .., Z_6^{(r)}$, whereas the final output transformation uses four 16-bit key subblocks $Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)}$ where $r = 1, 2, .., 8$. The encryption algorithm of IDEA can be described as :

$$IDEA(X) = A(P(E^{(8)}(E^{(7)}(\ldots(E^{(1)}(X_1^{(0)}, X_2^{(0)}, X_3^{(0)}, X_4^{(0)}))))), Z_\gamma) = Y.$$

where $E^{(r)} = P(H(MA(L(A(X^{(r-1)}, Z_\alpha)), Z_\beta))) = X^{(r)}$ is the r^{th} round and $X^{(r-1)} = (X_1^{(r-1)}, X_2^{(r-1)}, X_3^{(r-1)}, X_4^{(r-1)})$, $Z_\alpha = (Z_1^{(r)}, Z_2^{(r)}, Z_3^{(r)}, Z_4^{(r)})$, $Z_\beta = (Z_5^{(r)}, Z_6^{(r)})$, $Z_\gamma = (Z_1^{(9)}, Z_2^{(9)}, Z_3^{(9)}, Z_4^{(9)})$,$P(X) = (X_1, X_3, X_2, X_4)$, $A(X, Z) = X \otimes Z = (X_1 \odot Z_1, X_2 \boxplus Z_2, X_3 \boxplus Z_3, X_4 \odot Z_4)$, $L(X) = (X_1 \oplus X_3, X_2 \oplus X_4)$, $H(X, T) = (X_1 \oplus T_2, X_2 \oplus T_1, X_3 \oplus T_2, X_4 \oplus T_1)$, and $MA(P_1, P_2, Z_1, Z_2) = (C_1, C_2)$ given by

$$\begin{aligned} C_2 &= ((K_1 \odot P_1) \boxplus P_2) \odot K_2. \\ C_1 &= (K_1 \odot P_1) \boxplus C_2. \end{aligned}$$

3.2.2 Key Schedule and Decryption Algorithm

Key schedule algorithm is exactly same as the algorithm explained in section 3.1.2. Similar to PES, IDEA uses the algorithm used for the encryption in its decryption. For the decryption process of IDEA (Figure 4), the ciphertext $Y = (Y_1, Y_2, Y_3, Y_4)$ taken as an input and the decryption key subblocks $K_i^{(r)}$ derived from the encryption key subblocks $Z_i^{(r)}$. The decryption key subblocks $K_i^{(r)}$ is computed as:

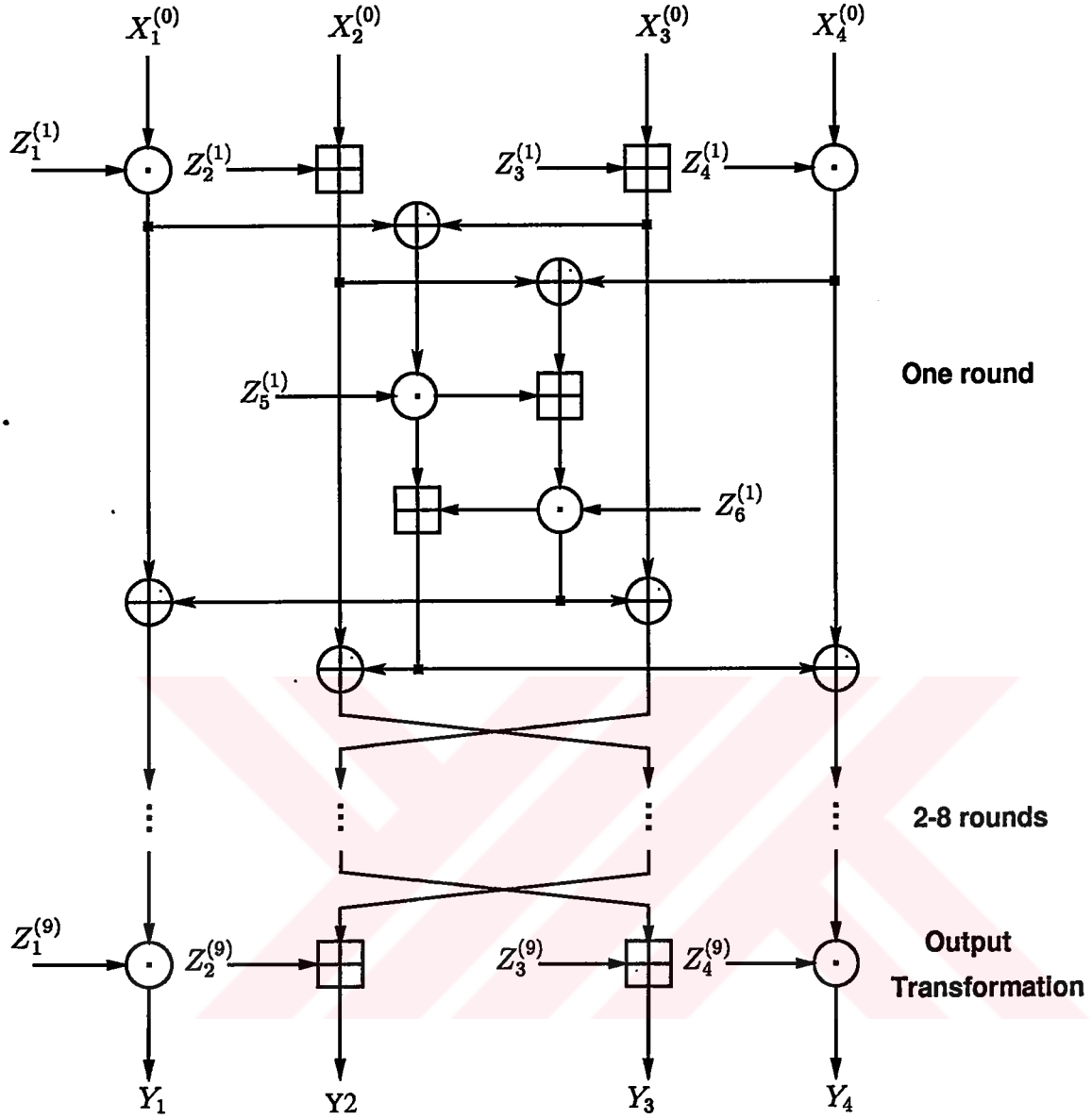


Figure 4 Computational graph for the encryption process of the IDEA cipher

For $r = 2, 3, .., 8$,
$\left(K_1^{(r)}, K_2^{(r)}, K_3^{(r)}, K_4^{(r)}\right) = \left(Z_1^{(10-r)}, -Z_3^{(10-r)}, -Z_2^{(10-r)}, Z_4^{((10-r))^{-1}}\right)$;
For $r = 1$ and 9,
$\left(K_1^{(r)}, K_2^{(r)}, K_3^{(r)}, K_4^{(r)}\right) = \left(Z_1^{(10-r)}, -Z_2^{(10-r)}, -Z_3^{(10-r)}, Z_4^{((10-r))^{-1}}\right)$;
For $r = 1, 2, .., 8$,
$\left(K_5^{(r)}, K_6^{(r)}\right) = \left(Z_5^{(r)}, Z_6^{(r)}\right)$;

where Z^{-1} denotes the multiplication inverse (modulo $2^{16} + 1$) of Z , i.e., $Z \odot Z^{-1} = 1$ and $-Z$ denotes the additive (modulo 2^{16}) of Z, i.e., $-Z \boxplus Z = 0$.

3.3 Security of IDEA

The designers of IDEA claimed that the required confusion for IDEA achieved by the interaction of the mixing operations from different algebraic groups ($\odot, \otimes$ and $\boxplus$) which have been described in the first section of this chapter. In [12] and [14] this interaction was discussed by using the concept of isotopism of quasigroups and polynomial expressions.

Definition 3.3.1 ([12],[14]) *Let S be a non-empty set and let $*$ denote an operation from pairs (a, b) of elements of S to an element $a * b$ of S. Then $(S, *)$ is said to be a* quasigroup *if, for all a and $b \in S$, the equation $a * x = b$ and $y * a = b$ both have exactly one solution in S.*

Definition 3.3.2 ([12],[14]) *The quasigroups $(S_1, *_1)$ and $(S_2, *_2)$ are said to be isotopic if there are bijective mappings $\theta, \phi, \psi : S_1 \to S_2$, such that,*

$$\theta(x) *_2 \phi(y) = \psi(x *_1 y) \text{ for all } x \text{ and } y \text{ in } S_1.$$

For the following theorem, let $(\mathbb{Z}_2^n, \oplus)$ denote the group of n-tuples over $\mathbb{Z}_2$ under the bitwise exclusive-OR operation.

Theorem 3.3.3 ([12],[14])*For $n \in \{1, 2, 4, 8, 16\}$:*

1) The quasigroups $(\mathbb{Z}_2^n, \oplus)$ and $(\mathbb{Z}_{2^n}, +)$ are not isotopic for $n \geq 2$.

2) The quasigroups $(\mathbb{Z}_2^n, \oplus)$ *and* $(\mathbb{Z}_{2^n+1}^*, .)$ *are not isotopic for* $n \geq 2$.

3) The triple (θ, ϕ, ψ) *of bijections from* $\mathbb{Z}_{2^n+1}^*$ *to* $\mathbb{Z}_{2^n}$ *is an isotopism of* $(\mathbb{Z}_{2^n+1}^*, .)$ *onto* $(\mathbb{Z}_{2^n}, +)$ *if and only if there exist* c_1 *and* c_2 *in* $\mathbb{Z}_{2^n}$ *and a generator* α *of the cyclic group* $\mathbb{Z}_{2^n+1}^*$ *such that, for all* x *in* $\mathbb{Z}_{2^n+1}^*$,

$$\theta(x) - c_1 = \phi(x) - c_2 = \psi(x) - (c_1 + c_2) = log_\alpha(x),$$

i.e., any isotopism between these groups is essentially the discrete logarithm. Moreover, when $n \geq 2$, *none of three bijections in an isotopism* (θ, ϕ, ψ) *from* $\mathbb{Z}_{2^n+1}^*$ *onto* $\mathbb{Z}_{2^n}$ *can be direct mapping d defined in section 3.1.1 .*

Proof: ([12],[14])

3.3.1 Polynomial expressions for multiplication and addition

The relation between multiplication modulo $2^n + 1$ and addition modulo 2^n are established by the direct mapping d and its inverse d^{-1} defined in (3.1). For the multiplication modulo $2^n + 1$, the function $g : \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ defined by

$$g(x, y) = d[d^{-1}(x).d^{-1}(y)) \bmod (2^n + 1)] \text{ for all } x \text{ and } y \in \mathbb{Z}_{2^n}.$$

For the addition modulo 2^n , the function $l : \mathbb{Z}_{2^n+1} \times \mathbb{Z}_{2^n+1} \to \mathbb{Z}_{2^n+1}$ defined by

$$l(x, y) = \begin{cases} d^{-1}[d(x) + d(y) \mod (2^n)] & \text{for all } x \text{ and } y \in \mathbb{Z}_{2^n+1} \\ 0 & \text{otherwise} \end{cases}$$

Example 3.3.4 ([12],[14]) *When* $n = 1$, $l(x, y) = 2xy$ *mod 3 for all* x *and* y *in* $\mathbb{Z}_3$ *and* $g(x, y) = x + y + 1$ *mod 2 for all* x *and* y *in* $\mathbb{Z}_2$.

In [12],[14] for the discussion of the nonlinearity of the function l and g over the field $\mathbb{Z}_{2^n+1}$ and over the ring $\mathbb{Z}_{2^n}$, respectively the following theorems are given:

Theorem 3.3.5 ([12],[14]) *For $n \in \{1,2,4,8,16\}$: For every a in $\mathbb{Z}_{2^n+1} - \{0,2^n\}$, the function $l(a,y)$ is a polynomial in y over field $\mathbb{Z}_{2^n+1}$ with degree 2^n-1. Similarly, for every a in $\mathbb{Z}_{2^n+1} - \{0,2^n\}$, the function $l(x,a)$ is a polynomial in x over $\mathbb{Z}_{2^n+1}$ with degree 2^n-1.*

Proof: ([12],[14])

Example 3.3.6 ([12],[14]) *For $n=2$, the function $l(x,y)$ over $\mathbb{Z}_5$ induced by addition modulo 4 is*

$$l(x,y) = 3(x^3y^2 + x^2y^3) + 3(x^3y + xy^3) + 2x^2y^2 + 4(x^2y + xy^2).$$

Theorem 3.3.7 ([12],[14]) *If $n \in \{1,2,4,8,16\}$, then, for every $a \in \mathbb{Z}_{2^n} - \{0,1\}$, the function $g(a,x)$ cannot be written as a polynomial in x over the ring $\mathbb{Z}_{2^n}$.*

Proof: ([12],[14])

The next section starts with two provable security features for IDEA, which are completely taken from [12],[14].

3.3.2 Security Features of IDEA

In this section, we state provable security features of the IDEA cipher.

1- Confusion

The confusion required for a secure cipher is achieved in the IDEA cipher by mixing three incompatible group operations. In the computational graph of the encryption process for IDEA, the three different group operations are so arranged that *the output of an operation of one type is never used as the input to an operation of the same type.*

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

The three operations are incompatible in the sense that:

1. No pair of the 3 operations satisfies a "distributive" law. For instance, for the operations $\odot$ and $\boxplus$, there exists a,b, and c in $\mathbb{Z}_2^{16}$, such that,

$$a \boxplus (b \odot c) \neq (a \boxplus b) \odot (a \boxplus c).$$

For example, when $a = b = c = 1 = (0, 0, \ldots, 0, 1) \in \mathbb{Z}_2^{16}$, the left side of the above inequality is $2 = (0, 0, \ldots, 0, 1, 0)$, while the right side equals $4 = (0, 0, \ldots.0, 1, 0, 0)$.

2. No pair of the 3 operations satisfies a "generalized associative" law. For instance, for the operations $\boxplus$ and $\oplus$, there exists a,b, and c in $\mathbb{Z}_2^{16}, \mathbb{Z}_2^{16}$,

$$a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c.$$

For example, when $a = b = c = 1 = (0, 0, \ldots, 0, 1)$, the left side of the above inequality is $1 = (0, 0, \ldots, 0, 0, 1)$, while the right side equals $3 = (0, 0, \ldots.0, 0, 1, 1)$. Thus, one cannot arbitrarily change the order of operations to simplify analysis.

3. The 3 operation are connected by the direct mapping d and its inverse,which inhibits isotopisms as shown in Theorem 3.3.3. The cryptographic significance of this fact, if there were an isotopism between two operations, then one could replace one operation with the other by applying bijective mappings on the inputs and on the output. It follows from Theorem 3.3.3 that $(\mathbb{Z}_2^{16}, \odot)$ and $(\mathbb{Z}_2^{16}, \oplus)$ are not isotopic and that $(\mathbb{Z}_2^{16}, \boxplus)$ and $(\mathbb{Z}_2^{16}, \oplus)$ are not isotopic. The isotopism from $(\mathbb{Z}_2^{16}, \odot)$ onto $(\mathbb{Z}_2^{16}, \boxplus)$ is essentially the discrete logarithm, which , as shown in Theorem 3.3.3, cannot be the direct mapping d. Moreover, the discrete logarithm is generally considered to be a "complex" function.

4. Under the direct mapping d and its inverse d^{-1}, it is possible to consider the operations $\odot$ and $\boxplus$ as acting on the same set (either in the ring $\mathbb{Z}_{2^n}$ or in the field $\mathbb{Z}_{2^n+1}$). However, by doing so, we must analyze some highly non-linear functions in the sense that multiplication modulo $2^{16}+1$, which is a bilinear function over $\mathbb{Z}_{2^{16}+1}$, correspond to a non-polynomial function over $\mathbb{Z}_{2^{16}}$, as shown Theorem 3.3.7 . Similarly, addition modulo 2^{16}, which is an affine function in each argument over $\mathbb{Z}_{2^{16}}$, correspond to a two variable polynomial of degree $2^{16}-1$ in each variable over $\mathbb{Z}_{2^{16}}$, as shown in Theorem 3.3.5.

2- Diffusion

A check by direct computation has shown that the round function is "complete", i.e., that each output bit of the first round depends on every bit of the plaintext and on every bit of the key used for that round. This diffusion is provided in the IDEA cipher by the transformation called the multiplication-addition (MA) structure whose computational graph is shown in Figure 5. The MA structure transforms two 16 bit subblocks controlled by two 16 bit key subblocks. This structure has the following properties:

- for any choice of the key subblocks Z_5 and Z_6, $MA(.,.,Z_5,Z_6)$ is an invertible transformation; for any choice of U_1 and U_2 , $MA(U_1,U_2,.,.)$ is also an invertible transformation;
- this structure has a "complete diffusion" effect in the sense that each output subblock depends on every input subblock, and

- this structure uses the least number of operations (four) required to achieve such complete diffusion. (pp 33, Lemma 2 in [14])

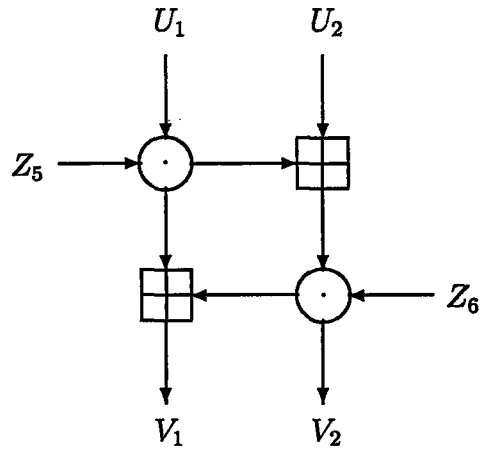


Figure 5 Computational graph of the MA structure

3- Resistance against Differential Cryptanalysis

In [14] it is shown that for a suitable chosen difference, IDEA(m) is a Markov cipher for $m = 8, 16, 32, 64$ where $m = 4n$ is the length of the plaintext and n is the length of each subblocks. In addition to this, three classes of highly probable differentials of the IDEA cipher is determined and according to these, its immunity against differential cryptanalysis is analyzed and concluded that the IDEA cipher is secure against a differential cryptanalysis attack after only 4 of its 8 rounds.

In the next chapter we shall investigate the nonlinearity of the mixing operations by viewing them on the vector space $\mathbb{Z}_2^n$, MA structure and slightly changed version of that structure. In [1] the diffusion properties of PES and

IDEA was investigated by considering two criteria and their test procedures given there and concluded that these block ciphers attain the required diffusion after only the first of their 8 rounds. Moreover, it was observed that MA structure indeed provides that diffusion and it requires at least 4 operation for the *complete diffusion* as it is stated in above,[12] and [14].

In the open literature, Meier [22] showed that the operations $\odot$ and $\boxplus$ satisfy a partial distributive law with a certain probability and presented an attack for 2-rounds of IDEA using that property. In [8] a large classes of weak keys was found for IDEA, which is the only attack published in literature. This is certainly better than exhaustive search on 128-bit key space and it was claimed that with a slight modification of the key schedule of IDEA, the problem of weak keys can be eliminated. Borst, Knudsen and Rijmen [5] presented two attacks on a reduced number of rounds of IDEA. One of them uses differential-linear attack for 3-rounds of IDEA and the other attack uses truncated differentials to analyze 3.5-rounds (3-rounds and an output transformation) of IDEA. Borst [6] described an attack on 3-rounds of IDEA using differential and linear cryptanalysis techniques.

3.3.3 Low-High algorithm for multiplication

We shall use the following lemma that is given for the implementation issues of IDEA and PES in [12] and [14] :

Lemma 3.3.8 (Low-High algorithm for $\odot$) *Let a, b be two n-bit non-zero integers in $\mathbb{Z}_{2^n+1}$, then*

$$ab \bmod (2^n+1) = \begin{cases} (ab \bmod 2^n) - (ab \text{ div}) (2^n) & \text{if } (ab \bmod 2^n) \geq (ab \text{ div } 2^n) \\ (ab \bmod 2^n) - (ab \text{ div } 2^n) + 2^n + 1 & \text{if } (ab \bmod 2^n) < (ab \text{ div } 2^n) \end{cases}$$

where (ab div 2^n) denotes the quotient when ab is divided by 2^n.

Proof: ([12],[14])

CHAPTER 4

THE NONLINEARITY PROPERTIES OF MA AND RMA STRUCTURES

As it is mentioned in the section 3.3.2, we have observed that one of the point on the security of IDEA is the fact that for $n = 2, 4, 8, 16$

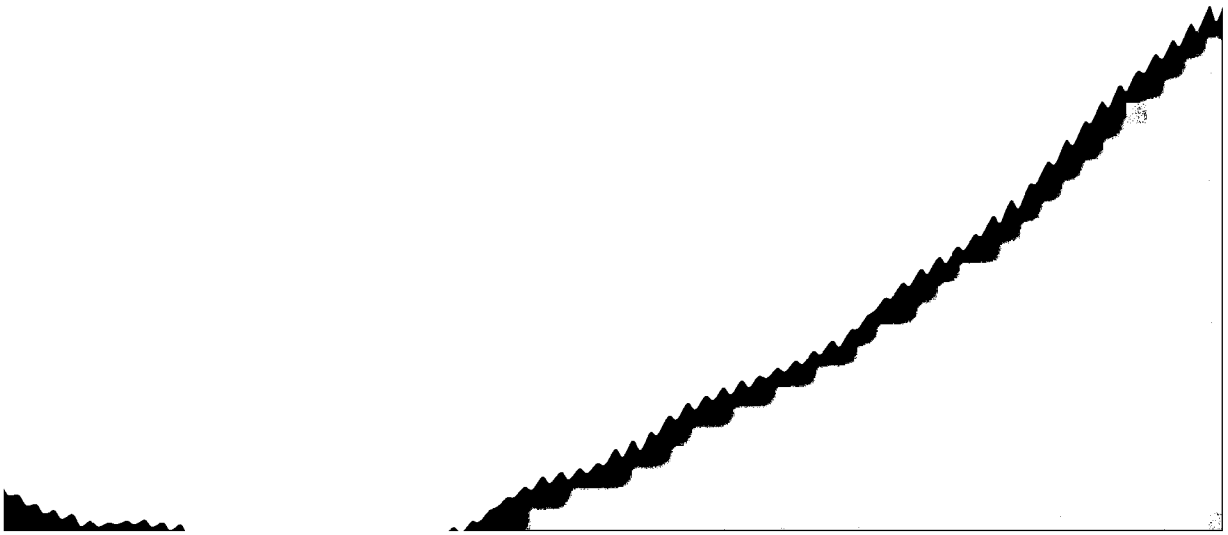
(i) for each $a \in \mathbb{Z}_{2^n+1} \backslash \{0, 2^n\}$, $l(x, a)$ is a polynomial in x over the ring $\mathbb{Z}_{2^n+1}$ with degree $2^n - 1$.

(ii) for each $a \in \mathbb{Z}_{2^n} \backslash \{0, 2^n\}$, $g(x, a)$ can not be written as a polynomial in x over the ring $\mathbb{Z}_{2^n}$, where

$$l(x,y) = \begin{cases} d^{-1}[d(x) + d(y) \mod (2^n)] & \text{for all } x \text{ and } y \in \mathbb{Z}_{2^n+1} \\ 0 & \text{otherwise} \end{cases}$$

$$g(x,y) = d[d^{-1}(x).d^{-1}(y)) \mod (2^n + 1)] \text{ for all } x \text{ and } y \text{ in } \mathbb{Z}_{2^n}.$$

The authors, due to the properties above,viewed these operations highly "nonlinear" as it has been explained in the section 3.3.2. One of each operations $\oplus$ and $\boxplus$ used in IDEA's encryption and decryption algorithm is a bilinear operation $E : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ such that for every given $k \in Z_2^n$ $E(., k)$ is an invertible vector function from $\mathbb{Z}_2^n \to \mathbb{Z}_2^n$ for $n = 2, 4, 8, 16$. For each k, one can look at the nonlinearity of $E(., k)$ over $\mathbb{Z}_2^n$. Our aim in this thesis is to view these operations on the corresponding vector spaces $\mathbb{Z}_2^n (\cong \mathbb{Z}_{2^n+1}^*)$ and $\mathbb{Z}_2^n (\cong \mathbb{Z}_{2^n})$ respectively, and discuss the nonlinearity of them in the sense of



Nyberg [21]. Of course, we know that for a reliable block cipher system, the nonlinearity of such structures is important to achieve the immunity for the linear attacks.

We have carried out some computations on the linearities of the operations $\odot, \oplus$ and $\boxplus$. According to our computational experiments on computers, for $n = 4$ and $n = 8$ the Nyberg's nonlinearity of the function $\tilde{f} : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ associated to $f(a, x) = a + x \mod (2^n)$, which denoted by $N(\tilde{f})$ becomes zero $\forall a \in \mathbb{Z}_{2^n}$. In fact, $\forall n \geq 1$, for a given integer a and $\forall x \in \mathbb{Z}_{2^n}$, the right-end component $\tilde{f}_0$ of the vector function $\tilde{f}(v(a), v(x))$, is $a_0 + x_0 \mod (2)$ where $v(a) = (a_{n-1}, a_{n-2}, \ldots, a_0)$, $v(x) = (x_{n-1}, x_{n-2}, \ldots, x_0)$ and $\tilde{f} = (\tilde{f}_{n-1}, \tilde{f}_{n-2}, \ldots, \tilde{f}_0)$. This relation is sufficient to make $N(\tilde{f})$ zero. In addition, $\forall a \in \mathbb{Z}_{2^n}$ each component of the function $\tilde{h}(a, x), \tilde{h}_i = a_i + x_i \mod (2)$ is an affine function $\forall i = 0, 1, \ldots$,n -1 and therefore we have

Proposition 4.0.9 *For $n \geq 1$, $N(\tilde{f}(v(a), .)$ and $N(\tilde{h}(v(a), .))$ equal to zero $\forall a \in \mathbb{Z}_{2^n}$.*

For the multiplication operation $\odot$, $\forall a \in \mathbb{Z}_{2^n}, N(\tilde{g}(v(a), .))$'s values are illustrated in Table 1 for n=4 and in Figure 6 for n=8. When n=4 and a=0, 1, 2, 8, 9, 15 $N(\tilde{g}(v(a), .)) = 0$ (see Table 1). Also when n=8 and a=0, 1, 2, 128, 129 and 255 $N(\tilde{g}(v(a), .)) = 0$ (see Figure 6).

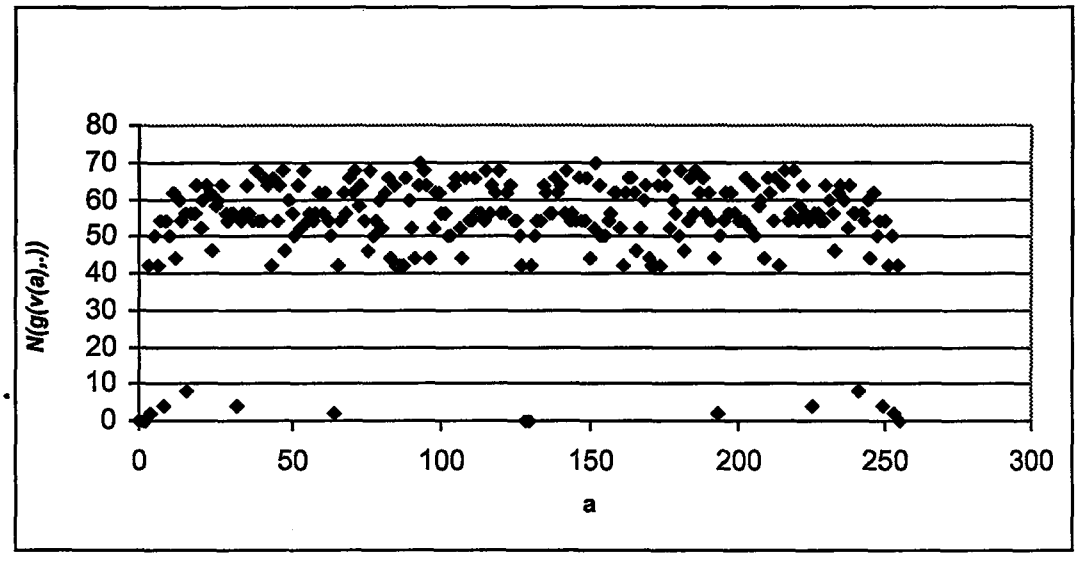


Figure 6 For n=8 $N(\tilde{g}(v(a),.))$'s values

Table 1 For n=4 $N(\tilde{g}(v(a),.))$'s values

a	$N(\tilde{g}(v(a),.))$	a	$N(\tilde{g}(v(a),.))$
0	0	8	0
1	0	9	0
2	0	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	0

From these observations, we reach the following generalization:

Theorem 4.0.10 *For $n \geq 2$, The Nyberg's nonlinearity of the vector function $\tilde{g}(v(a), v(x))$, namely $N(\tilde{g})((v(a),.) = 0$ for $a = 0, 1, 2, 2^{n-1}, 2^{n-1}+1, 2^n - 1$.*

Proof: $\forall a, x \in \mathbb{Z}_{2^n}$, let us denote $\tilde{g}(v(a), v(x))$ by
$\tilde{g}_a(v(x)) = v(a) \odot v(x) = v(g(a,x)) = v(d(d^{-1}(a) \cdot d^{-1}(x) \bmod (2^n+1)))$,
and take $d^{-1}(x) = \tilde{x} = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^i$ and $x = \sum_{i=0}^{n-1} x_i \cdot 2^i$.

We shall prove the theorem for each $a = 0,\ 1,\ 2,\ 2^{n-1},\ 2^{n-1}+1,\ 2^n-1$.
For $\mathbf{a = 0}$, we claim the right-end component of $\tilde{g}_0(v(x))$ is given by $x_0 + 1$ which is affine in $x_0, \ldots, x_{n-1}$ and thus $N(\tilde{g}_0) = 0$.

$g(0,x) = d(2^n \cdot \tilde{x} \bmod (2^n+1)) \quad \forall x \in \mathbb{Z}_{2^n}$
$2^n \cdot \tilde{x} \bmod (2^n) = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^{i+n} \bmod (2^n) = 0$ and
$2^n \cdot \tilde{x} \text{ div } (2^n) = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^{i+n} \text{ div } (2^n) = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^i = \tilde{x} \neq 0.$

From these equation we have $2^n \cdot \tilde{x} \bmod (2^n) < 2^n \cdot \tilde{x} \text{ div } (2^n)$ by using lemma 3.3.8,
$2^n \cdot \tilde{x} \bmod (2^n+1) = 2^n \cdot \tilde{x} \bmod (2^n) - 2^n \cdot \tilde{x} \text{ div } (2^n) + 2^n + 1.$

Case 1: For $x = 0$, In that case, $\tilde{x} = 2^n, \tilde{x}_n = 1$
and $\tilde{x}_i = 0\ \forall i \in \{0, 1, \ldots, n-1\}$ and we have
$2^n \cdot \tilde{x} \bmod (2^n+1) = 2^n + 1 - 2^n = 1$. Hence $\tilde{g}_0(v(0)) = (0, \ldots, 0, 1)$.

Case 2: For $x \neq 0$, $\tilde{x} = x$ and

$$\begin{aligned} 2^n \cdot x \bmod (2^n+1) &= 2^n + 1 - x \\ &= \sum_{i=0}^{n-1} 2^i + 2 - \sum_{i=0}^{n-1} x_i \cdot 2^i = 2 + \sum_{i=0}^{n-1} (1 - x_i) = 2 + \bar{x}. \end{aligned}$$

where $\bar{x}$ is the bitwise complement of x. Let $(s_{n-1}, \ldots, s_0)$ be components of $\tilde{g}_0(v(x))$, namely $\tilde{g}_0(v(x)) = (s_{n-1}(v(x)), \ldots, s_0(v(x)))$ where $s_i : \mathbb{Z}_2^n \to \mathbb{Z}_2$ for

$i = 0, \ldots, n-1$. For this case, the right-end component $s_0(v(x))$ of $\tilde{g}_0(v(x))$ is $1 + x_0$. The same result also holds for the *Case 1*, because in that case $x_0 = 0$ and $s_0(v(x)) = 1 = 1 + x_0$. Hence $s_0(v(x)) = 1 + x_0$ is affine for both cases, and therefore $N(\tilde{g}_0) = 0$.

For a = 1, we have

$g(1, x) = d(1 \cdot \tilde{x} \bmod (2^n + 1)) \quad \forall x \in \mathbb{Z}_{2^n}$

$1 \cdot \tilde{x} \bmod (2^n) = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^i \bmod (2^n) = \tilde{x}$ and $1 \cdot \tilde{x}$ div $(2^n) = \tilde{x}_n$.

Case 1: For $x = 0$,(i.e. $\tilde{x} = 2^n$) and we obtain

$0 = 1 \cdot \tilde{x} \bmod (2^n) < 1 \cdot \tilde{x}$ div $(2^n) = 1$. From lemma 3.3.8, we observe

$g(1, x) = d(1 \cdot \tilde{x} \bmod (2^n) - 1 \cdot \tilde{x}$ div $(2^n) + 2^n + 1) = d(2^n) = 0$

and $\tilde{g}_1(v(0)) = (0, \ldots, 0)$.

Case 2: For $x \neq 0$,

$0 = 1 \cdot \tilde{x}$ div $(2^n) < 1 \cdot \tilde{x} \bmod (2^n) = x$ since $\tilde{x} = x$ and $\tilde{x}_n = 0$. Again by lemma 3.3.8, $g(1, x) = d(x) = x$.

From the above cases, $\tilde{g}_1(v(x)) = (x_{n-1}, \ldots, x_0) = v(x)$ and each component of $\tilde{g}_1$ is a linear function in $x_0, \ldots, x_{n-1}$ and thus $N(\tilde{g}_1) = 0$.

For a = 2,

$g(2, x) = d(2 \cdot \tilde{x} \bmod (2^n + 1)) \quad \forall x \in \mathbb{Z}_{2^n}$

$2 \cdot \tilde{x} \bmod (2^n) = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^{i+1} \bmod (2^n) = \sum_{i=0}^{n-2} \tilde{x}_i \cdot 2^{i+1}$

and $2 \cdot \tilde{x}$ div $(2^n) = \sum_{i=0}^{n} \tilde{x}_i \cdot 2^{i+1}$ div $(2^n) = \tilde{x}_{n-1} + 2 \cdot \tilde{x}_n$.

Case 1: For $x = 0$ (i.e. $\tilde{x}_n = 1$ and $\tilde{x}_i = 0 \quad \forall i \in \{0, 1, \ldots, n-1\}$),
By the inequality $2 \cdot \tilde{x}$ div $(2^n) > 2 \cdot \tilde{x}$ mod (2^n) and lemma 3.3.8,
$g(2, x) = d(2 \cdot \tilde{x} \text{ mod } (2^n) - 2 \cdot \tilde{x} \text{ div } (2^n) + 2^n + 1) = d(2^n + 1 - 2) = 2^n - 1$
and $\tilde{g}_2(v(x)) = (1, 1, \ldots, 1)$.

Case 2: Let us consider the case $\tilde{x} \neq 0, x_i = 0$ for $i \in \{0, 1, \ldots, n-2\}$
.and $x_{n-1} = 1$. Then one can observe
$2 \cdot x$ div $(2^n) > 2 \cdot x$ mod (2^n) and conclude that $g(2, x) = d(2^n) = 0$ by lemma
3.3.8. Hence $\tilde{g}_2(v(x)) = (0, 0, \ldots, 0)$.

Case 3: For $x_{n-1} = 0$ and $\exists i \in \{0, 1, \ldots, n-2\}$ such that $x_i \neq 0$.
In this case one gets $\tilde{g}_2(v(x)) = (x_{n-2}, x_{n-3}, \ldots, x_0, 0)$.

Case 4: For $x_{n-1} = 1$ and $x_0 = 0$ and $x_i \neq 0$
for at least one $i \in \{1, \ldots, n-2\}$. Then we consider an index t such that
$x_0 = x_1 = \ldots = x_{t-1} = 0$ and $x_t = 1$, we have $2 \cdot x$ mod $(2^n) = \sum_{i=0}^{n-2} x_i \cdot 2^{i+1}$
and $2 \cdot \tilde{x}$ div $(2^n) = x_{n-1} = 1$ by using lemma 3.3.8 , we obtain

$2 \cdot \tilde{x}$ mod $(2^n + 1) = \sum_{i=0}^{n-2} x_i \cdot 2^{i+1} - 1$.

Thus one can check that $\tilde{g}_2(v(x)) = (x_{n-2}, x_{n-3}, \ldots, x_{t+1}, 0, 1, 1, \ldots, 1)$.

Case 5: For $x_{n-1} = 1$ and $x_0 = 1$,
Similar to *Case 4*, $\tilde{g}_2(v(x)) = (x_{n-2}, x_{n-3}, \ldots, x_1, 0, 1)$.

We can summarize all these as follows:

$$\tilde{g}_2(v(x)) = \begin{cases} (1,1,\ldots,1) & v(x) = (0,0,\ldots,0) \\ (0,0,\ldots,0) & v(x) = (1,0,\ldots,0) \\ (x_{n-2},x_{n-3},\ldots,x_0,0) & v(x) = (0,x_{n-2},\ldots,x_0) \\ (x_{n-2},\ldots,x_{t+1},0,1,1,\ldots,1) & v(x) = (1,x_{n-2},\ldots,x_{t+1},1,0,\ldots,0) \\ (x_{n-2},\ldots,x_1,0,1) & v(x) = (1,x_{n-2},\ldots,x_1,1) \end{cases}$$

As before, if $(p_{n-1},\ldots,p_0)$ denotes the components of $\tilde{g}_2 : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, it is not difficult to see from the above that $p_0(x_{n-1},x_{n-2},\ldots,x_0) + p_1(x_{n-1},x_{n-2},\ldots,x_0) = x_0$ is affine, and thus $N(\tilde{g}_2) = 0$.

For $a = 2^{n-1}$,

$g(2^{n-1},x) = d(2^{n-1}\cdot\tilde{x} \bmod (2^n+1)) \quad \forall x \in Z_{2^n}$

$2^{n-1}\cdot\tilde{x} \bmod (2^n) = \sum_{i=0}^{n}\tilde{x}_i\cdot 2^{n+i-1} \bmod (2^n) = \tilde{x}_0\cdot 2^{n-1}$

and $2^{n-1}\cdot\tilde{x}$ div $(2^n) = \sum_{i=0}^{n}\tilde{x}_i\cdot 2^{n+i-1}$ div $(2^n) = \sum_{i=1}^{n}\tilde{x}_i\cdot 2^{i-1}$.

To compute $\tilde{g}_{2^{n-1}} = (d_{n-1},\ldots,d_0)$, where for $i \in \{0,1,\ldots,n-1\}$ $d_i(x) : \mathbb{Z}_{2^n} \to \mathbb{Z}_2$ is the component function of $\tilde{g}_{2^{n-1}}$. We need to look at several cases:

Case 1: For $x \neq 0$ (i.e. $\tilde{x} = x$) and $x_0 \neq 0$,

$2^{n-1} = x_0\cdot 2^{n-1} = 2^{n-1}\cdot x \bmod (2^n) \geq 2^{n-1}\cdot x$ div $(2^n) = \sum_{i=1}^{n-1}x_i\cdot 2^{i-1}$.

Then by lemma 3.3.8 ,

$2^{n-1}\cdot x \bmod (2^n+1) = \sum_{i=0}^{n-2}2^i + 1 - \sum_{i=1}^{n-1}x_i\cdot 2^{i-1} = 1 + \sum_{i=1}^{n-1}(1-x_i)\cdot 2^{i-1}$

and from that equation the right-end component of the $\tilde{g}_{2^{n-1}}(v(x))$,

$d_0(v(x)) = (2-x_1) \bmod (2) = x_1 \;\forall\; x \in \mathbb{Z}_{2^n}$.

Case 2: For $x \neq 0$ and $x_0 = 0$,

$\sum_{i=1}^{n-1} x_i \cdot 2^{i-1} = 2^{n-1} \cdot x \text{ div } (2^n) > 2^{n-1} \cdot x \bmod (2^n) = 2^{n-1} \cdot x_0 = 0.$

From Lemma 3.3.8,

$2^{n-1} \cdot x \bmod (2^n + 1) = 2^n + 1 - (\sum_{i=1}^{n-1} x_i \cdot 2^{i-1})$

$= \sum_{i=0}^{n-2} 2^i + 2 - \sum_{i=1}^{n-1} x_i \cdot 2^{i-1} = 2 + \sum_{i=1}^{n-1}(1 - x_i) \cdot 2^{i-1} + 2^{n-1}$

Hence, the right-end component of the $\tilde{g}_{2^{n-1}}(v(x))$,

$d_0(v(x)) = 1 - x_1 \bmod (2) = 1 + x_1 \ \forall \ x \ \in \mathbb{Z}_{2^n}.$

Case 3: For $x = 0\,(\tilde{x} = 2^n)$,

By lemma 3.3.8, $2^{n-1} \cdot \tilde{x} \bmod (2^n + 1) = 2^n + 1 - 2^{n-1} = 2^{n-1} + 1$

since $2^{n-1} \cdot \tilde{x} \bmod (2^n) = 0$ and $2^{n-1} \cdot \tilde{x} \text{ div } (2^n) = 2^{n-1}$

and we have $\tilde{g}_{2^{n-1}}(v(0)) = (1, 0, \ldots, 0, 1)$. With this we computed explicitly all the values $\tilde{g}_{2^{n-1}}(v(x)) \ \forall x \in \mathbb{Z}_{2^n}$ where $v(x) = (x_{n-1}, x_{n-2}, \ldots, x_0) \in \mathbb{Z}_2^n$. Now we leave the reader to check that $d_0(v(x)) = x_0 + x_1 + 1 \, \forall x \in \mathbb{Z}_{2^n}$ where $\tilde{g}_{2^{n-1}}(v(x)) = (d_{n-1}, \ldots, d_0)$. This gives immediately $N(\tilde{g}_{2^{n-1}}) = 0$.

For $\mathbf{a = 2^{n-1} + 1}$, we have

$g(2^{n-1} + 1, x) = d((2^{n-1} + 1) \cdot \tilde{x} \bmod (2^n + 1)) \quad \forall x \in \mathbb{Z}_{2^n}$

$$\begin{aligned}(2^{n-1} + 1) \cdot \tilde{x} \bmod (2^n) &= (\sum_{i=0}^{n} \tilde{x}_i \cdot 2^{n+i-1} + \sum_{i=0}^{n} \tilde{x}_i \cdot 2^i) \bmod (2^n) \\ &= \tilde{x}_0 + 2 \cdot \tilde{x}_1 + \ldots + (\tilde{x}_0 + \tilde{x}_{n-1}) \cdot 2^{n-1}\end{aligned}$$

$$\begin{aligned}(2^{n-1} + 1) \cdot \tilde{x} \text{ div } (2^n) &= (\sum_{i=0}^{n} \tilde{x}_i \cdot 2^{n+i-1} + \sum_{i=0}^{n} \tilde{x}_i \cdot 2^i) \text{ div } (2^n) \\ &= \tilde{x}_1 + \tilde{x}_n + \tilde{x}_2 \cdot 2 + \ldots + \tilde{x}_n \cdot 2^{n-1}.\end{aligned}$$

Let $\tilde{g}_{2^{n-1}+1}(v(x)) = (t_{n-1}(v(x)), \ldots, t_0(v(x)))$. We claim that $t_0(v(x)) = t_0(x_{n-1}, \ldots, x_0) = x_0 + x_1$. This certainly gives $N(\tilde{g}_{2^{n-1}+1}) = 0$.

We look at several cases:

Case 1: For $x = 0$ and $\tilde{x}_n = 1$,
$2^{n-1}+1 = (2^{n-1}+1)\cdot\tilde{x} \text{ div } (2^n) > (2^{n-1}+1)\cdot\tilde{x} \text{ mod } (2^n) = 0.$
$(2^{n-1}+1)\cdot\tilde{x} \text{ mod } (2^n+1) = 2^n+1-(2^{n-1}+1) = 0$

since by lemma 3.3.8, $t_0(v(x)) = 0$.

Case 2: For $x \neq 0$ (i.e., $\tilde{x} = x$) and $x_0 = 0$,
$(2^{n-1}+1)\cdot x \text{ mod } (2^n) = 2\cdot x_1 + \ldots + x_{n-1}\cdot 2^{n-1}$

and $(2^{n-1}+1)\cdot x \text{ div } (2^n) = x_1 + x_2\cdot 2 + \ldots + x_{n-1}\cdot 2^{n-2}$. It is clear for $x_0 = 0$, $(2^{n-1}+1)\cdot x \text{ mod } (2^n) > (2^{n-1}+1)\cdot x \text{ div } (2^n)$ and in fact it follows the lemma 3.3.8, $(2^{n-1}+1)\cdot x \text{ mod } (2^n) - (2^{n-1}+1)\cdot x \text{ div } (2^n) =$
$(2^{n-1}+1)\cdot x \text{ div } (2^n) = x_1 + x_2\cdot 2 + \ldots + x_{n-1}\cdot 2^{n-2}$.

Thus for this case we get

$\tilde{g}_{2^{n-1}+1}(v(x)) = \tilde{g}_{2^{n-1}+1}(x_{n-1}, x_{n-2}, \ldots, 0) = (0, x_{n-1}, \ldots, x_1)$.

Now we look at the only remaining case:

Case 3: For $x \neq 0$ and $x_0 \neq 0$, i.e., $x_0 = 1$.
In that case, when $x_{n-1} = 0$, having
$(2^{n-1}+1)\cdot x \text{ mod } (2^n) > (2^{n-1}+1)\cdot x \text{ div } (2^n)$

and from lemma 3.3.8, we have

$$(2^{n-1}+1)\cdot x \text{ mod } (2^n+1) = (2^{n-1}+1)\cdot x \text{ mod } (2^n) - (2^{n-1}+1)\cdot x \text{ div } (2^n)$$
$$= 1 + x_1\cdot 2 + \ldots + x_{n-2}\cdot 2^{n-2} + 2^{n-1} - (x_1 + 2\cdot x_2 + \ldots + x_{n-2}\cdot 2^{n-3})$$
$$= 1 + 2\cdot t - t = 1 + t = 1 + x_1 + 2\cdot x_2 + \ldots + x_{n-2}\cdot 2^{n-3}.$$

where $t = (x_1 + 2 \cdot x_2 + \ldots + x_{n-2} \cdot 2^{n-3})$. This implies that $t_0(x_{n-1}, \ldots, x_0) = t_0(0, x_{n-2}, \ldots, x_1, 0) = 1 + x_1$.

Now when $x_{n-1} = 1$, it can be checked by induction that $(2^{n-1}+1) \cdot x \bmod (2^n) < (2^{n-1}+1) \cdot \tilde{x} \operatorname{div} (2^n)$. For this case,

$$(2^{n-1}+1) \cdot x \bmod (2^n) = 1 + 2 \cdot x_1 + \ldots + x_{n-2} + \ldots + x_{n-2} \cdot 2^{n-2}$$

and $(2^{n-1}+1) \cdot x \operatorname{div} (2^n) = 1 + x_1 + 2 \cdot x_2 + \ldots + x_{n-2} \cdot 2^{n-3} + 2^{n-2}$.

By using lemma 3.3.8 again we achieve $(2^{n-1}+1) \cdot x \bmod (2^n+1) = 2^n + 1 + (2^{n-1}+1) \cdot x \bmod (2^n) - (2^{n-1}+1) \cdot x \operatorname{div} (2^n) = 2 + (1 + 2 + \ldots + 2^{n-1}) + 1 + 2 \cdot t - 1 - t - 2^{n-2} = 1 + x_1 + 2 \cdot x_2 + \ldots + x_{n-2} \cdot 2^{n-3} + 2^{n-2} + 2^{n-1}$ where t $=(x_1 + 2 \cdot x_2 + \ldots + x_{n-2} \cdot 2^{n-3})$. Thus $t_0(v(x)) = t_0(1, x_{n-2}, \ldots, x_1, 1) = (1 + x_1) \bmod 2$. It follows from all these calculation for three cases that $t_0(x_{n-1}, x_{n-2}, \ldots, x_0) = x_0 + x_1$. This means that $N(\tilde{g}_{2^n+1}) = 0$.

For a $= 2^n - 1$,

Using $2^n - 1 = -2 \bmod (2^n + 1)$, one can check that

$$\tilde{g}_{2^n-1}(v(x)) = v(d(2^n + 1 - d^{-1}(v^{-1}(\tilde{g}_2(v(x)))))))).$$

By using the results obtained for the case a=2 and the above observation, it is not hard to see that if $\tilde{g}_{2^n-1}(v(x)) = (s_{n-1}, \ldots, s_1, s_0)$, then $s_0(x_{n-1}, \ldots, x_0) + s_1(x_{n-1}, \ldots, x_0) = x_0 + 1$ which is affine, and for this reason $N(\tilde{g}_{2^n-1}) = 0$. Here are the cases:

Case 1: For $x = 0$,

$$\tilde{g}_{2^n-1}(v(0)) = v(d(2^n + 1 - d^{-1}(v^{-1}(\tilde{g}_2(v(0)))))) = v(d(2^n + 1 - d^{-1}(\sum_{i=0}^{n-1} 2^i)))$$

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

$$= \quad v(d(2^n + 1 - (2^n - 1))) = v(2) = (0, \ldots, 1, 0).$$

Case 2: For $x \neq 0$ and $x_i = 0$ for $i \in \{0, 1, \ldots, n-2\}$ and $x_{n-1} = 1$,

$$\begin{aligned} \tilde{g}_{2^n-1}((1, 0, \ldots, 0)) &= v(d(2^n + 1 - d^{-1}(0))) \\ &= v(d(2^n + 1 - 2^n)) = v(1) = (0, 0, \ldots, 1). \end{aligned}$$

Case 3: For $x_{n-1} = 0$ and $\exists\, i \in \{0, 1, \ldots, n-2\}$ such that $x_i \neq 0$.

In this case similarly,

$$\begin{aligned} \tilde{g}_{2^n-1}((0, x_{n-2}, \ldots, x_0)) &= v(d(2^n + 1 - 2 \cdot x_0 - 2^2 \cdot x_1 - \ldots - x_{n-2} \cdot 2^{n-1})) \\ &= (, \ldots, x_0, 1). \end{aligned}$$

Case 4: For $x_{n-1} = 1$ and $x_0 = 0$ and for at least one $i \in \{1, \ldots, n-2\}$ such that $x_i \neq 0$,

Then we consider an index t such that $x_0 = x_1 = \ldots = x_{t-1} = 0$ and $x_t = 1$,we have $\tilde{g}_{2^n-1}((1, x_{n-2}, x_{n-3}, \ldots, x_{t+1}, 1, 0, \ldots, 0)) = v(d(2^n + 1 - (2^{n-1} \cdot x_{n-2} + 2^{n-2} \cdot x_{n-3} + \ldots + 2^{t+2} \cdot x_{t+1} + 2^t + \ldots + 2 + 1))) = (., \ldots, 1, 0)$.

Case 5: For $x_{n-1} = 1$ and $x_0 = 1$,

$\tilde{g}_{2^n-1}((1, x_{n-2}, x_{n-3}, \ldots, x_1, 1)) = v(d(2^n + 1 - (2^{n-1} \cdot x_{n-2} + 2^{n-2} \cdot x_{n-3} + \ldots + 2^2 \cdot x_1 + 1))) = (., \ldots, 0, 0)$.

It can be now check easily that $s_0(v(x)) + s_1(v(x)) = 1 + x_0$ for all $x \in \mathbb{Z}_{2^n}$.

This finishes the proof of the theorem. □

Remark 4.0.11 *The case $a = 0$ in the previous theorem was stated in [8].*

Now we are going to look at the nonlinearity of the combinations of the operation $\boxplus$ and $\odot$. We start with the nonlinearity of the functions

$\tilde{k}_{a,b}(x) = (v(a) \odot v(x)) \boxplus v(b)$ and $\tilde{l}_{a,b}(x) = (v(a) \boxplus v(x)) \odot v(b)$, respectively. Our computational experiments in the case of $n = 2, 4$ show that $N(\tilde{k}_{a,b}) = 0$, when a = 0, 1, 2, 2^{n-1}, $2^{n-1}+1$, $2^n - 1$, and $N(\tilde{l}_{a,b}) = 0$, when b= 0, 1, 2, 2^{n-1}, $2^{n-1}+1$, $2^n - 1$. These calculations demonstrate that the linearity of the operation $\odot$ is carried out over the operation $\boxplus$. In addition, In the case of n= 4, out of 256 possible pairs of (a, b),

(i) 112 pairs make $N(\tilde{k}_{a,b})$ and $N(\tilde{l}_{a,b})$ equal to 2.

(ii) 144 pairs make $N(\tilde{k}_{a,b})$ and $N(\tilde{l}_{a,b})$ equal to 0.

The combinations of the operations $\boxplus$ and $\odot$ successively used in a special part of each round of IDEA, called as Multiplication- Addition (MA) structure (see Figure 7).This can be viewed as a transformation $MA : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \times \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n \times \mathbb{Z}_2^n$ given by

$$MA(P_1, P_2, K_1, K_2) = (C_1, C_2)$$

where (P_1, P_2) are two n-bit input subblocks, C_1, C_2 are two n-bit output subblocks and K_1, K_2 are two n-bit key subblocks . According to designers of IDEA, this transformation leads the block cipher IDEA to achieve the required diffusion. In the light of the above observations, it is natural to discuss the nonlinearity properties of MA structure in the sense of Nyberg. For this calculation, one can consider for each fixed $K = (K_1, K_2) \in \mathbb{Z}_2^{2n}$ transformation $M : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n \times \mathbb{Z}_2^n$ given by

$$M(P_1P_2, K_1, K_2) = (C_1C_2)$$

where $C_2 = ((K_1 \odot P_1) \boxplus P_2) \odot K_2$

and $C_1 = (K_1 \odot P_1) \boxplus C_2$.

For each $K = (K_1, K_2)$ this gives us to look at this transformation as an $2n \times 2n$ S-Box (a permutation over $\mathbb{Z}_2^{2n}$) . Now in addition to that transformation, we are going to discuss a slightly different one which we call Reverse MA, and denoted it by $RM : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n \times \mathbb{Z}_2^n$. In fact, for each fixed $K = (K_1, K_2) \in \mathbb{Z}_2^{2n}$,

$$RM(P_1P_2, K_1, K_2) = M(K_1K_2, P_1, P_2) = (C_1C_2)$$

where $C_2 = ((K_1 \odot P_1) \boxplus K_2) \odot P_2$

and $C_1 = (K_1 \odot P_1) \boxplus C_2$.

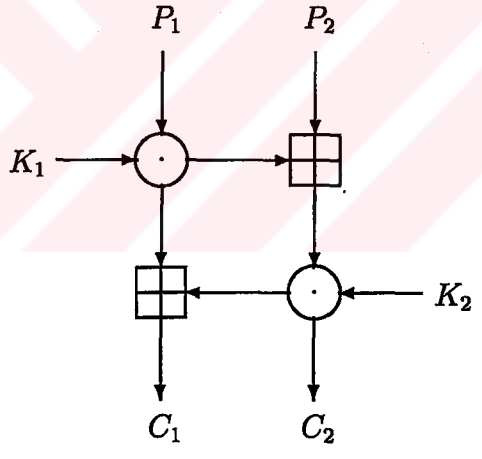


Figure 7 Computational graph of the MA structure

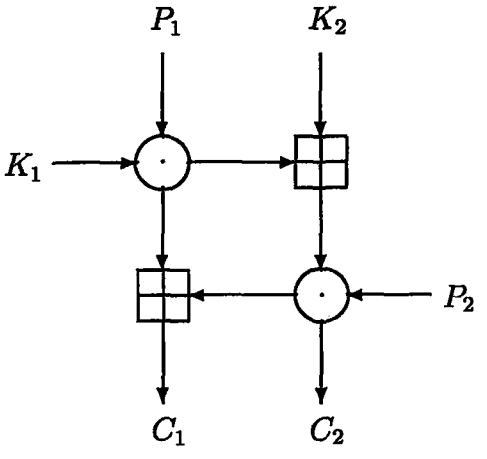


Figure 8 Computational graph of the RMA structure

From the Figure 8, it can be observed that the only difference between MA and RMA structures is the interchanging the position of P_2 and K_2.

We consider the case n=2 and 4 to measure the nonlinearity of the transformation M and RM for several cases. For n = 2 and for all possible key pairs, $N(M)$ and $N(RM)$ are zero.

For n=4, $N(M) = 0$ when $v^{-1}(K_1) = 0, 1, 8, 9$ or $v^{-1}(K_2) = 0, 1, 2, 8, 9, 15$. For $v^{-1}(K_1) = 2$ and 15, $N(M) = 32$ which is the smallest values among the other nonzero values. When n=4, the histogram of the nonlinearity values for all possible key values for this can be found in Figure 9. MA structure actually consists of the function $\tilde{k}_{a,b}$ and $\tilde{l}_{a,b}$, and we have observed the effects of the linearity of the multiplication operation $\odot$ over these combinations.

When we think the case Reverse MA for n=4, we obtain Figure 10 which shows the occurrence of the nonlinearity values of RM. For all key $v^{-1}(K_2)$ and

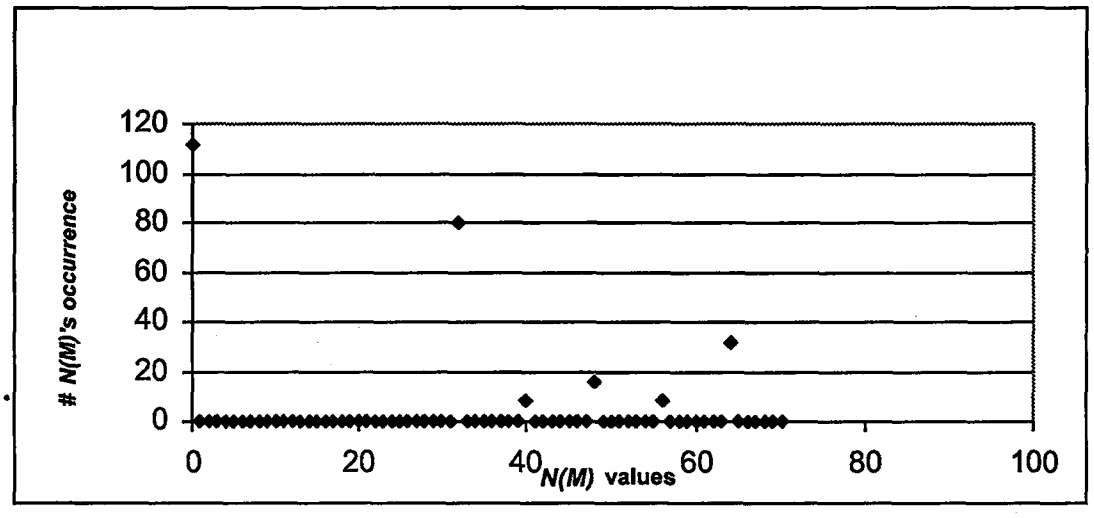


Figure 9 For n=4 Histogram of $N(M)$ over all 256 values of the transformation M.

$v^{-1}(K_1) = 0, 1, 8$ and 9 $N(RM)$ equals to 0. For K_2, we see that linearity of the multiplication does not affect the nonlinearity of Reverse MA. However, the first key K_1 is still included in Reverse MA structure by the multiplication as in MA structure and this multiplication gives zero values for $N(RM)$. In IDEA, each round contains one MA structure but one should consider the iteration of this structure due to existence 8 iterated rounds. Hence MA structure also iterates 8 times. For this reason, we also think of the composition of two MA ($M \circ M$), MA and Reverse MA ($M \circ RM$), Reverse MA and MA ($RM \circ M$) and two Reverse MA ($RM \circ RM$). Table 2 illustrates the occurrence of the nonlinearity values of these compositions for the case n = 2 and for all possible keys values. Here each composition is viewed as a transformation from $M : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n \times \mathbb{Z}_2^n$.

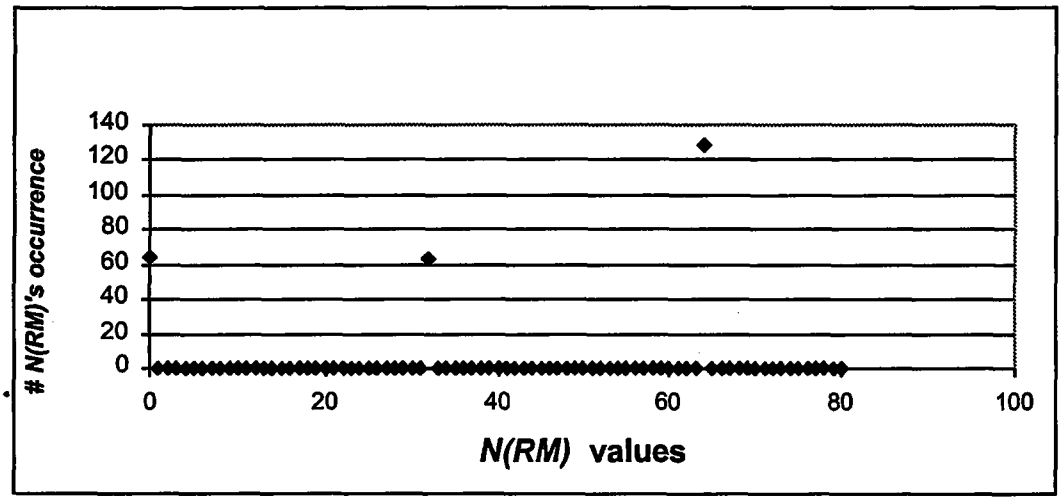


Figure 10 For n=4 Histogram of $N(RM)$ over all 256 values of the transformation RM

Table 2 For n=2 The occurrence of the nonlinearity of compositions' values

$N(\ldots)$	0	2	4
$\#N(M \circ M)$	224	0	32
$\#N(RM \circ RM)$	0	192	64
$\#N(RM \circ M)$	192	0	64
$\#N(M \circ RM)$	192	64	0

When n=4, for each composition, the nonlinearity of 2^{16} transformations can be computed for all key values. Due to our computation power we used some fixed set of key values for each combination and ended up computing the nonlinearity of the 2^{11} transformations. Figure 11,12,13 and 14 show the histogram of the nonlinearity values for these compositions.

From these figures, we conclude that the nonlinearity of the composition $M \circ M$ gives more zero values than other compositions; however, there are no

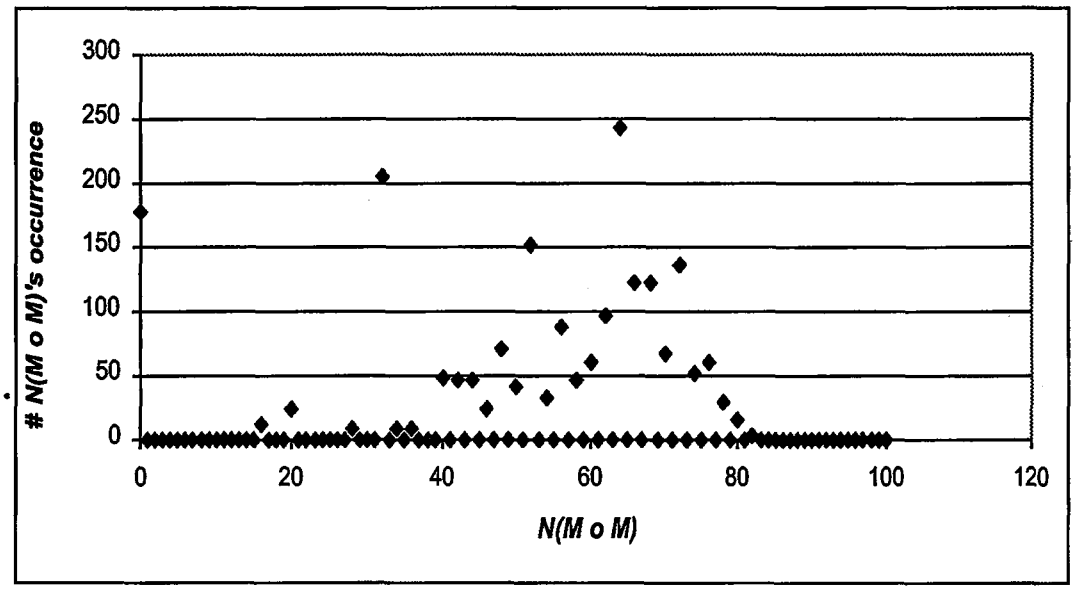


Figure 11 For n=4 Histogram of $N(M \circ M)$ over 2^{11} randomly chosen values of the transformation $M \circ M$.

zero values for $RM \circ RM$ and this composition attains the highest nonlinearity values among the other compositions (see Figure 14). On the other hand, $RM \circ M$ and $M \circ RM$ give better values than $M \circ M$.

If the nonlinearity of an n-block transformation is equal to zero, the nonzero linear combination of the its components is an affine function. So this transformation is not immunize to linear cryptanalysis. In fact, such transformation is immune against linear cryptanalysis if their nonlinearity is near to the integer 2^{n-1} as it has been explained in section 2.3.5. In this sense, transformations obtained from the composition $RM \circ RM$ are more resistant to linear cryptanalysis than the other compositions.

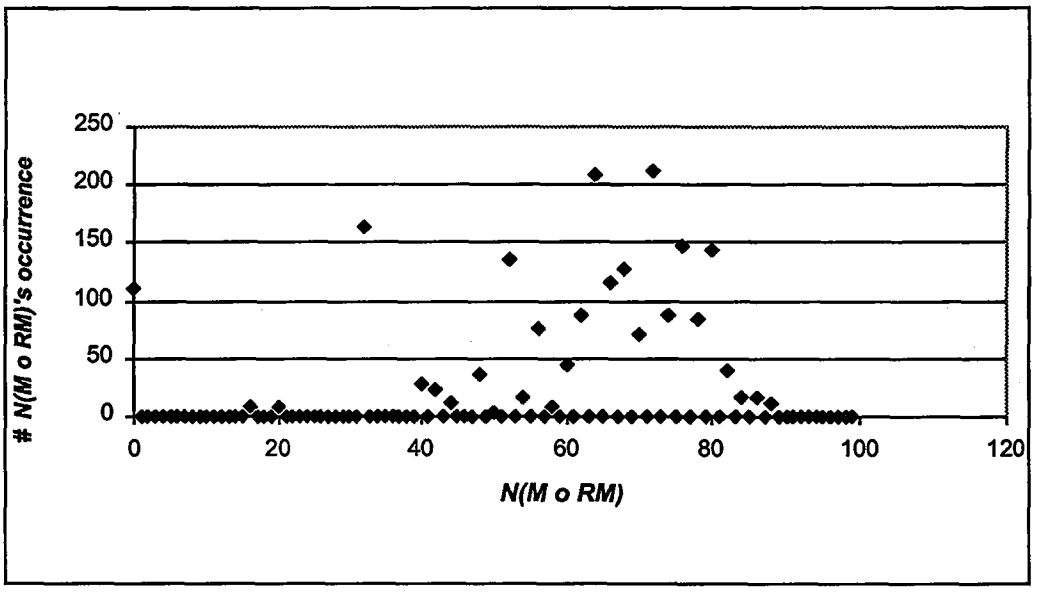


Figure 12 For n=4 Histogram of $N(M \circ RM)$ over 2^{11} randomly chosen values of the transformation $M \circ RM$.

Finally, the nonlinearities for the compositions $M \circ M \circ M$ (Figure 15), $RM \circ M \circ RM$ (Figure 16) , $M \circ RM \circ M$ (Figure 17) and $RM \circ RM \circ RM$ (Figure 18) are calculated when n=4. In this case, for each composition there are 2^{24} possible transformations for all key values. Again due to computation power we only consider 1280 of them and computed the corresponding nonlinearities. These calculations are shown in the Figure 15, 16, 17 and 18. As we can see $N(M \circ M \circ M)$ gives more zero values than others, and there are no zero values for $N(RM \circ RM \circ RM)$.

According to our computational experiments, instead of MA structure used in IDEA, we suggest the usage of new structure Reverse MA in IDEA encryption algorithm. Simply, this modification is minor and it does not change any other structure such as encryption, decryption and key schedule algorithm.

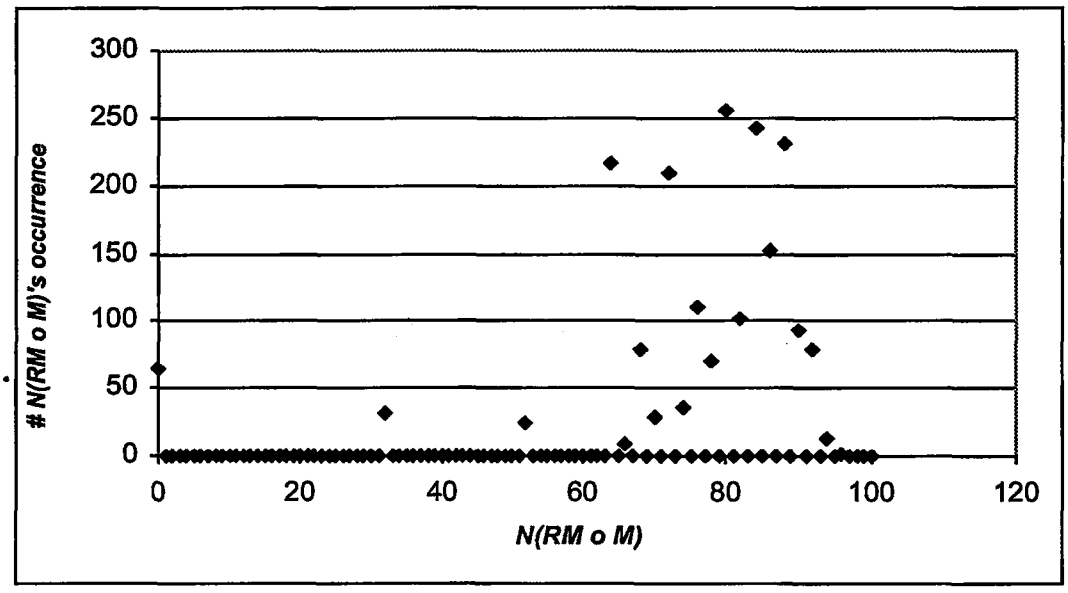


Figure 13 For n=4 Histogram of $N(RM \circ M)$ over 2^{11} randomly chosen values of the transformation $RM \circ M$.

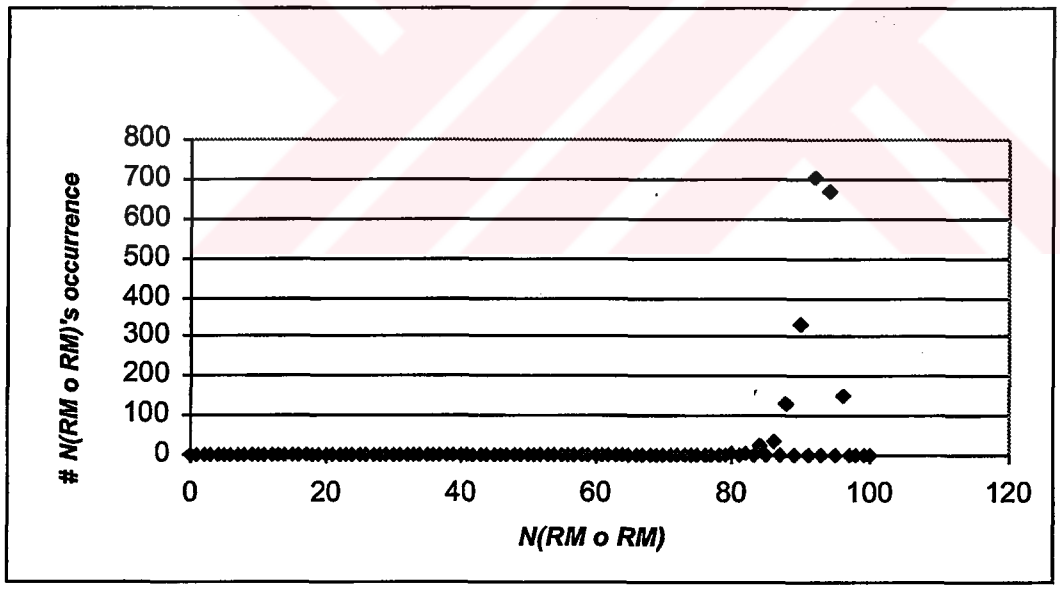


Figure 14 For n=4 Histogram of $N(RM \circ RM)$ over 2^{11} randomly chosen values of the transformation $RM \circ RM$.

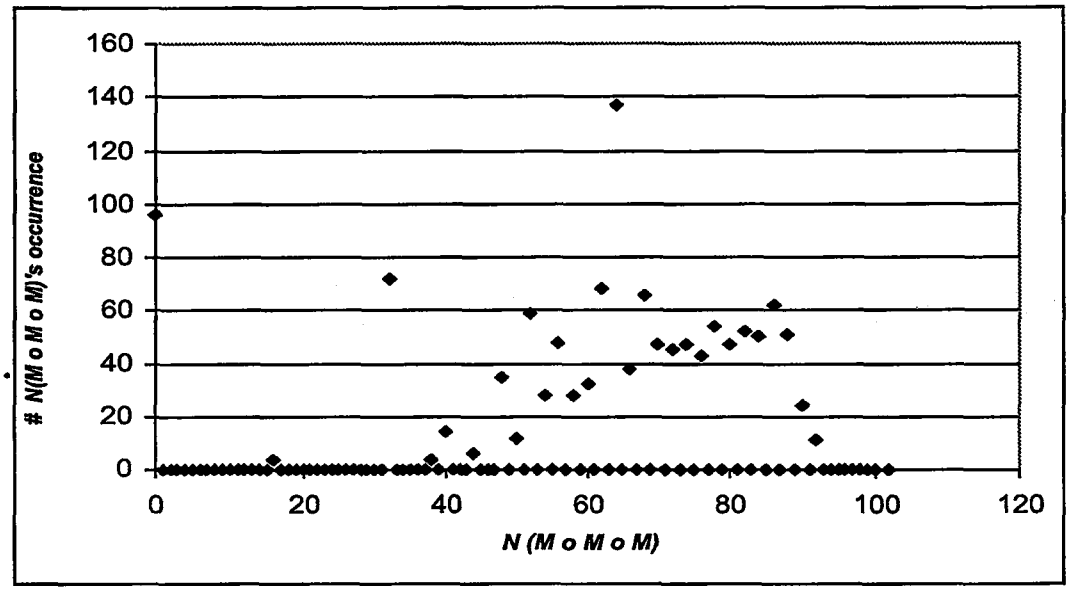


Figure 15 For n=4 Histogram of $N(M \circ M \circ M)$ over 1028 randomly chosen values of the transformation $M \circ M \circ M$.

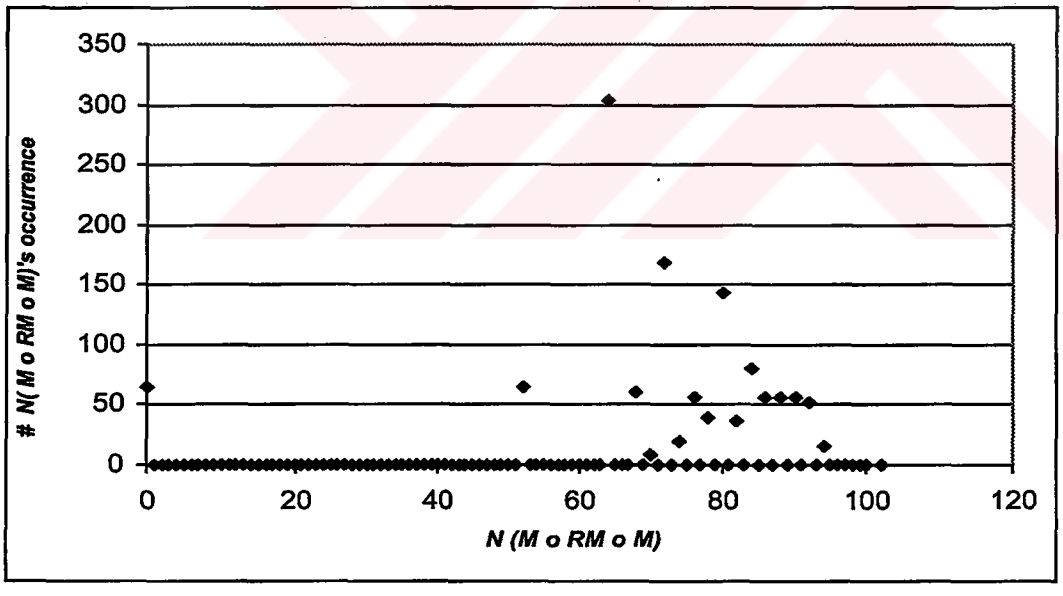


Figure 16 For n=4 Histogram of $N(M \circ RM \circ M)$ over 1028 randomly chosen values of the transformation $M \circ RM \circ M$.

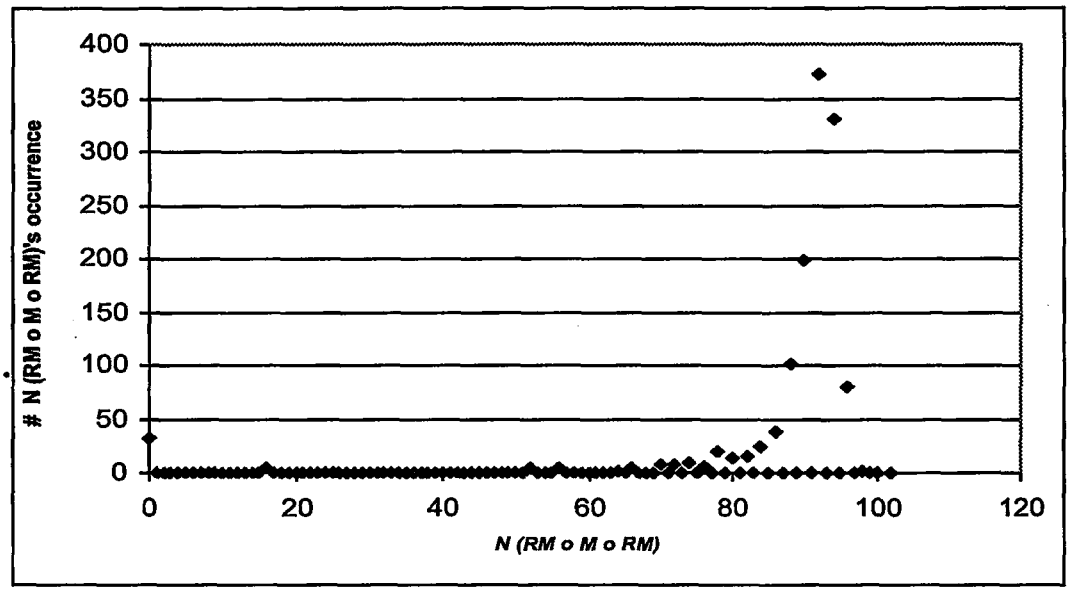


Figure 17 For n=4 Histogram of $N(RM \circ M \circ RM)$ over 1028 randomly chosen values of the transformation $RM \circ M \circ RM$.

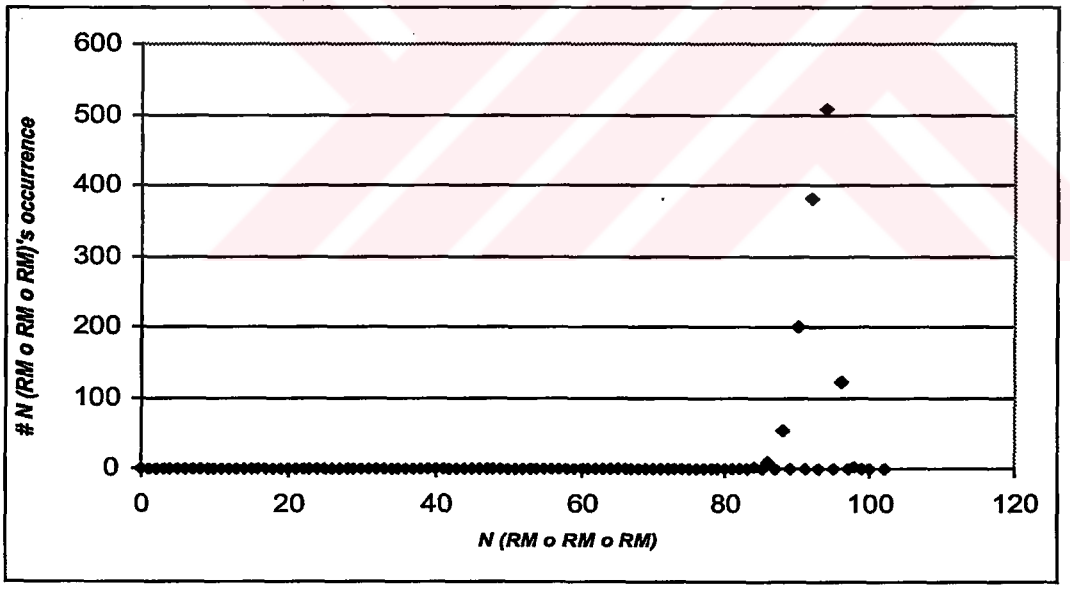


Figure 18 For n=4 Histogram of $N(RM \circ RM \circ RM)$ over 1028 randomly chosen values of the transformation $RM \circ RM \circ RM$.

We call this slightly modified version of IDEA having the Reverse MA structure as RIDEA (Reverse International Data Encryption Algorithm). The computational graph of RIDEA is shown in Figure 19. For its security one needs to check the diffusion properties. For this purpose, we use the Avalanche Weight Distribution (AWD), Avalanche criteria and their test procedures given in [1] to analyze the diffusion properties of our RIDEA. It is interesting to note that for 1-round RIDEA we obtain nearly the same average AWD curve (see Figure 21) as it was obtained for 1-round IDEA (see Figure 20). Similar to the avalanche_sum_array values of Average Avalanche curve of 1-round IDEA in [1] (see Figure 22), these values of 1-round RIDEA (see Figure 23) is nearly 50000 for each $i \in \{1, 2, \ldots, 64\}$ when the number of trials is 100000. As it is shown in [1] for that curve the maximum rate of change is within $\pm$ 0,7 % , but in our case it is within $\pm$ 0,1 %. From these observations we conclude that the required diffusion for RIDEA is provided at its first round. We believe that RMA structure is responsible for that diffusion.

The minor change applied to MA structure, Reverse MA structure not only provide required diffusion but also increase the nonlinearity of IDEA. This change is also minor change as in the case of PES, but it looks like it is an another step taken for reaching a more secure block cipher than IDEA.

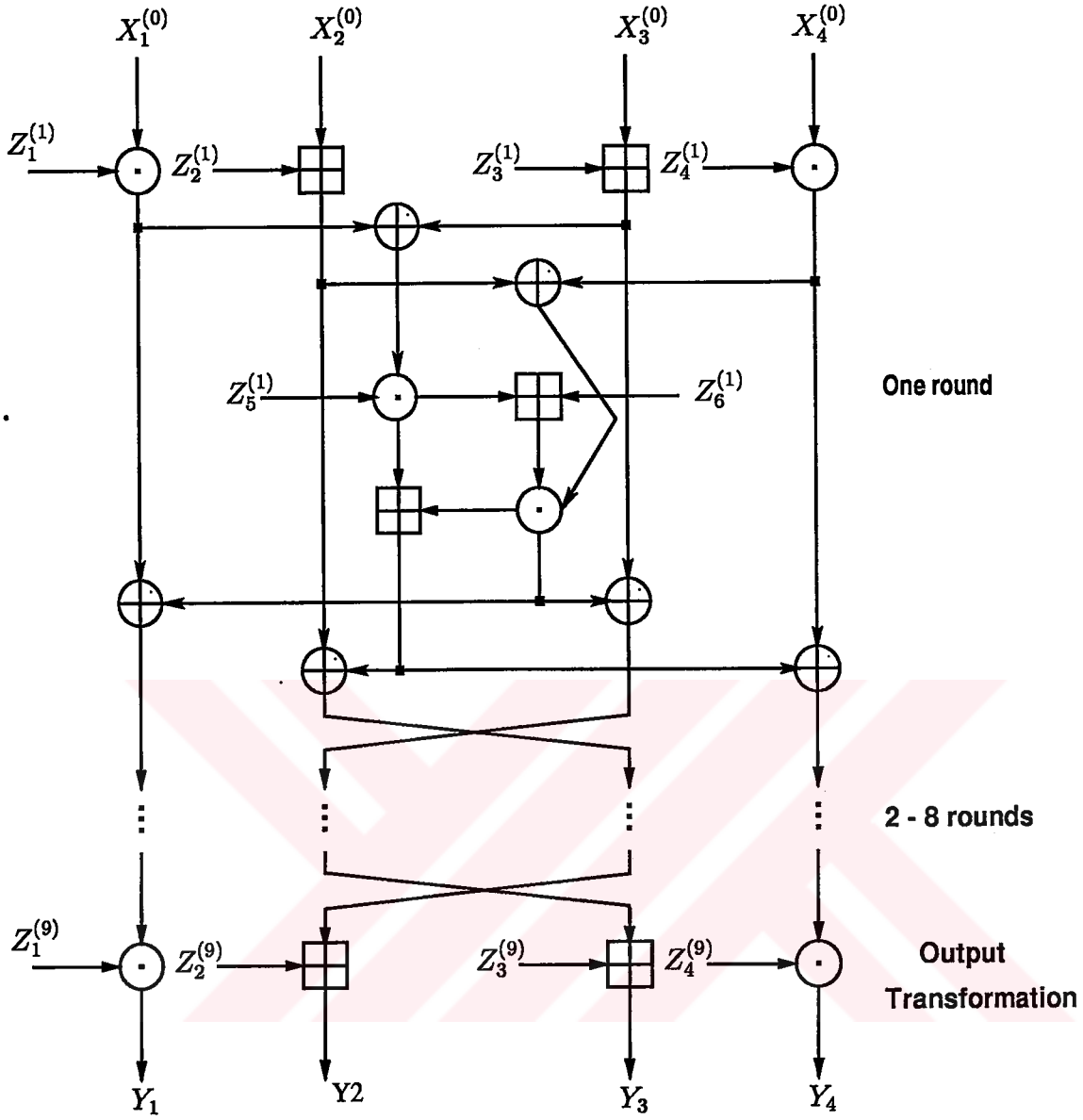


Figure 19 Computational graph for the encryption process of the RIDEA cipher

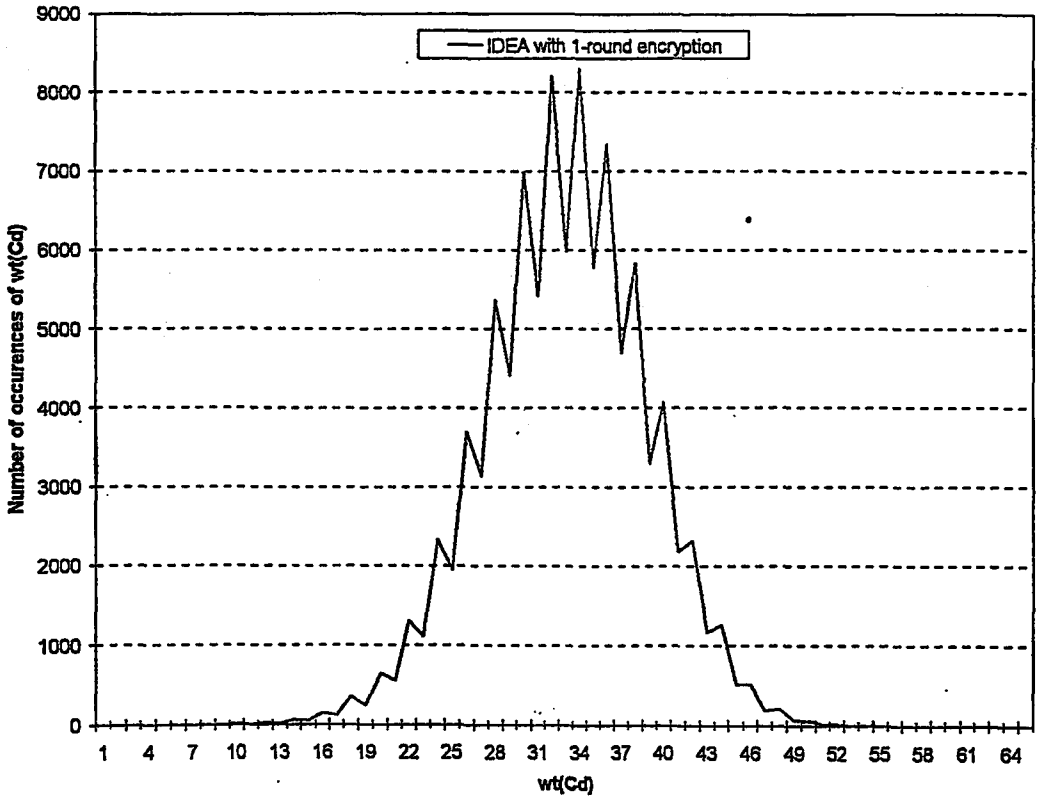


Figure 20 AWD curve for 1-round IDEA [1].

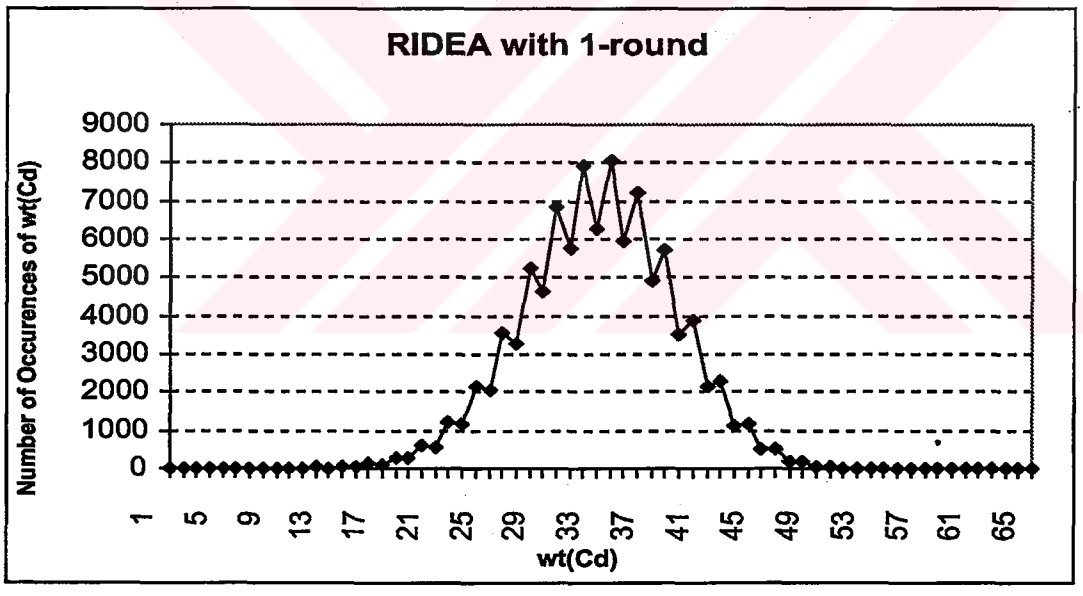


Figure 21 AWD curve for 1-round RIDEA.

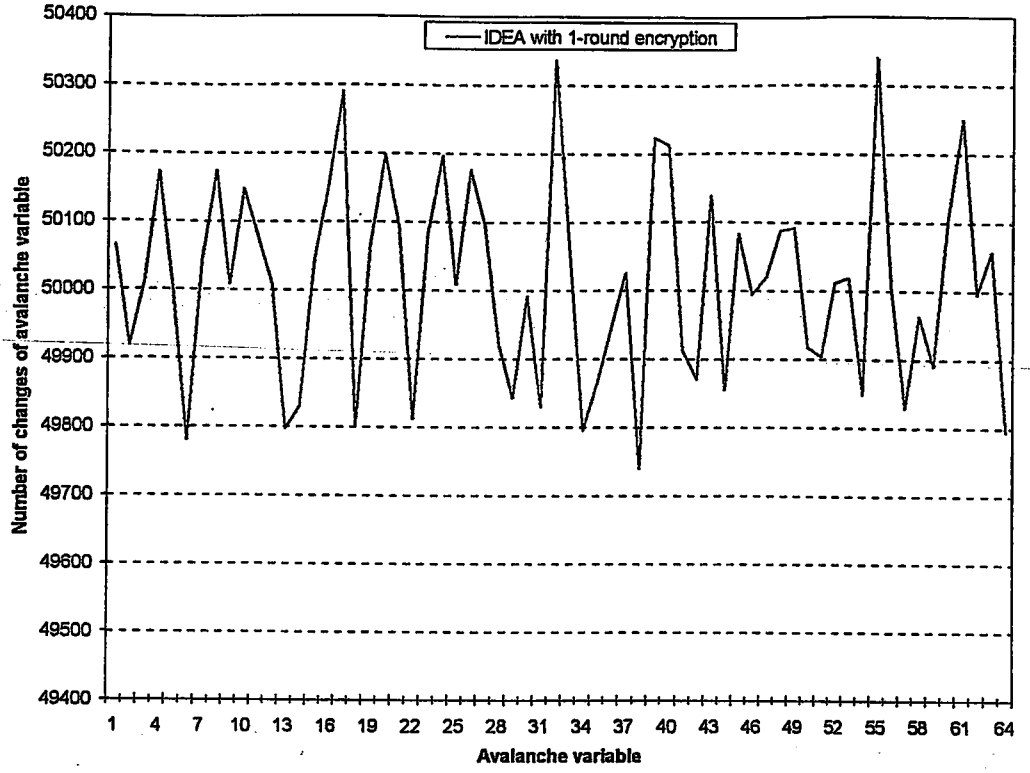


Figure 22 Average Avalanche curve of 1-round IDEA [1].

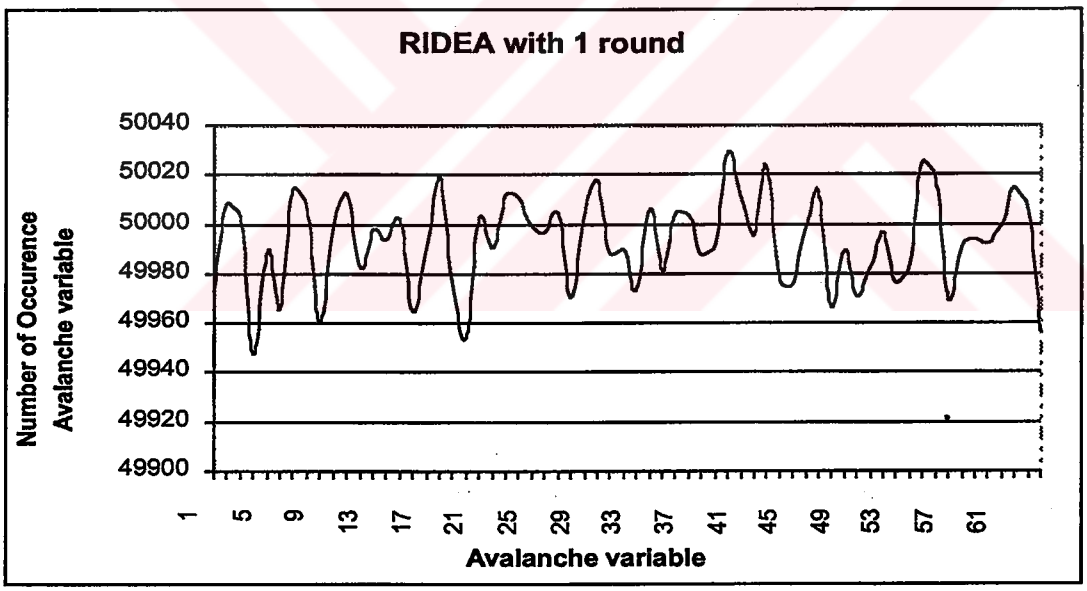


Figure 23 Average Avalanche curve of 1-round RIDEA values.

CHAPTER 5

CONCLUSION

We considered the MA structure used in block cipher system IDEA. We viewed the MA structure as a permutation on $\mathbb{Z}_2^n \times \mathbb{Z}_2^n$, and look at the nonlinearity properties of it in the sense of Nyberg/Matsui. We realized by slightly changing MA structure which we call RMA structure, we obtained far better nonlinearity properties. From this observation, we changed the MA structure in IDEA with RMA to obtain so-called RIDEA (Reverse IDEA) block cipher. This cipher has better nonlinear properties than IDEA, and keeps at least the same level diffusion property as in IDEA. Therefore, we propose in our thesis to change the block cipher IDEA with RIDEA to have more secure cipher.

In the future, as in [8], we hope to use Theorem 4.0.10 to produce some new classes of weak keys, by producing linear factors for an IDEA round. Moreover, we would like to study the differential and linear cryptanalysis of the RIDEA.

REFERENCES

[1] E. Aras, *Analysis of Security Criteria For Block Ciphers*, M.S Thesis, Department of Electrical and Electronics Engineering, METU, 1999.

[2] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Journal of Cryptology, Vol.4, No. 1,pp. 3-72, 1991.

[3] E. Biham and A. Shamir, *Differential Cryptanalysis of the Full 16-round DES*, Advances in Cryptology – CRYPTO '92, Springer Verlag, pp.487-496, 1992

[4] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[5] J. Borst, L.R. Knudsen and V. Rijmen, *Two Attacks on Reduced IDEA (Extended Abstract)*, Advances in Cryptology - EUROCRYPTO'97, Proceedings, Springer-Verlag, pp. 1–13, 1998.

[6] J.Borst, *Differential-Linear Cryptanalysis of IDEA*, ESAT-COSIC Technical Report 96-2.

[7] D. Coppersmith, *The Data Encryption Standard (DES) and its strentgh Aganist Attacks*, IBM J. Research and Development Vol. 38, No.3, pp. 243-250, May 1994.

[8] J.Daeman,R. Govaerts and J. Vandewalle, *Weak Keys for IDEA*, Advances

in Cryptology, Proc. EUROCRYPTO'93, LNCS 773, Springer-Verlag, pp. 224-231, 1994.

[9] B. Kaliski, M. Robshaw , *Linear Cryptanalysis Using Multiple Approximations*, CRYPTO'94, LNCS 839, page 26-38, (1994)

[10] L.R. Knudsen, *Truncated and higher order differentials*", In B. Preneel, editor, Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008, pages 196-211. Springer Verlag, 1995.

[11] T. Jakobsen and L.R. Knudsen, *The Interpolation Attack on Block Ciphers*, Fast Software Encryption, LNCS 1267, pp. 28-40. Springer Verlag, 1997.

[12] X. Lai and J. L. Massey, *A Proposal for a New Block Encryption Standard*, Advances in Cryptology - EUROCRYPTO'90, Proceedings, LNCS 473, pp. 389-404, Springer-Verlag, Berlin, 1991.

[13] X. Lai, J. L. Massey and S. Murphy, *Markov Cipher and Differential Cryptanalysis*, Advances in Cryptology - EUROCRYPTO'91. Lecture Notes in Computer Science 547, Springer Verlag.

[14] X. Lai, *On the design and security of block cipher*, ETH Series in Information Processing, V.1, Konstanz: Hartung-Gorre Verlag, 1992.

[15] X. Lai, *Higher order derivatives and differential cryptanalysis*, In Proc. "Symposium on Communication, Coding and Cryptography", in honor of James L. Massey on the occasion of his 60'th birthday, Feb. 10-13, 1994, Monte-Verita, Ascona, Switzerland, 1994.

[16] S.K. Langford and M.E. Hellman, *Differential- linear cryptanalysis*, Advances in Cryptography – CRYPTO '94 (LNCS 839), pp. 17-25,1994.

[17] M. Matsui and A.Yamagishi, *A New Method for Known Plaintext Attack of FEAL Cipher*, Lectures Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'92, pp. 81-91, 1992.

[18] M. Matsui , *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology — EUROCRYPT'93, 386 -397.

[19] M. Matsui , *The First Experimental Cryptanalysis of the Data Encryption Standard*, Advances in Cryptology — CRYPTO'94. 1-11

[20] K. Nyberg and L.R. Knudsen, *Provable security against differential cryptanalysis*, Advances in Cryptology – CRYPTO '92 pp. 566-574, 1992.

[21] K. Nyberg, *On the construction of highly nonlinear permutations*, In Extended Abstracts – EUROCRYPTO'92, pages 89-94, May 1992.

[22] W. Meier, *On the security of the IDEA Block Cipher*, Advances in Cryptology, Proc. EUROCRYPTO'93, LNCS 765, T. Helleseth, Ed.,Springer-Verlag,pp. 371-385, 1994.

[23] W. Meier and O. Staffelbach, *Nonlinearity criteria for cryptographic functions*, Proc. EUROCRYPTO'89, LNCS, Advances in Cryptology, 434 549-562, 1989.

[24] S. Murphy, *The Cryptanalysis of FEAL-4 with 20 Chosen Plaintexts.* Journal of Cryptography, No.3, 1990

[25] J. Pieprzyk, C. Charnes and J. Seberry, *Linear Approximation versus Nonlinearity* ,(ed Tavares,S), Selected Areas in Cryptography (SAC'94), Kingston, Ontario, 5-6 May,pp 82-90, 1994.

[26] M. Robshaw and L.R. Knudsen, *Non-Linear Approximations in Linear Cryptanalysis*, EUROCRYPTO'96, LNCS 1070, pp. 224-236, 1996.

[27] C. Shannon, *Communication Theory of Secrecy Systems.* Bell Systems Technical Journal, v28, Oct 1949, pp. 659-715.

[28] T. Shimoyama and T. Kaneko, *Quadratic Reation of S-box and Its Application to the Linear Attack of Full Round DES*, Advances in Cryptology, CRYPTO'98, LNCS 1462, pp. 200-211

[29] A. Webster and S. Tavares, *On the Design of S-Boxes*, Advances in Cryptology – CRYPTO '85 pp. 523-534, 1985.

[30] M. Weiener, *Efficient DES Key Search*, Proc. CRYPTO'93, 1993, published by Springer-Verlag.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ