ON THE AVALANCHE PROPERTIES OF MISTY1, KASUMI AND KASUMI-R

SEDAT AKLEYLEK

FEBRUARY 2008

ON THE AVALANCHE PROPERTIES OF MISTY1, KASUMI AND KASUMI-R

A THESİS SUBMİTTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEDAT AKLEYLEK

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

FEBRUARY 2008

Approval of the Graduate School of Applied Mathematics

Prof. Dr. Ersan AKYILDIZ
Director

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

Prof. Dr. Ferruh ÖZBUDAK
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.

Assoc. Prof. Dr. Melek Diker YÜCEL
Supervisor

Examining Committee Members

Prof. Dr. Ersan AKYILDIZ

Assoc. Prof. Dr. Melek Diker YÜCEL

Prof. Dr. Ferruh ÖZBUDAK

Assoc. Prof. Dr. Emrah ÇAKÇAK

Dr. Hamdi Murat YILDIRIM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name	: Sedat AKLEYLEK
Signature	:

ABSTRACT

ON THE AVALANCHE PROPERTIES OF MISTY1, KASUMI AND KASUMI-R

AKLEYLEK, Sedat
M.Sc., Deparment of Cryptography
Supervisor : Melek Diker YÜCEL

February 2008, 69 pages

The Global System for Mobile (GSM) Communication is the most widely used cellular technology. The privacy has been protected using some version of stream ciphers until the 3rd Generation of GSM. KASUMI, a block cipher, has been chosen as a standard algorithm in order to be used in 3rd Generation.

In this thesis, s-boxes of KASUMI, MISTY1 (former version of KASUMI) and RIJNDAEL (the Advanced Encryption Standard) are evaluated according to their linear approximation tables, XOR table distributions and satisfaction of the strict avalanche criterion (SAC). Then, the nonlinear part, FI function, of KASUMI and MISTY1 are investigated for SAC. A new FI function is defined by replacing both s-boxes of KASUMI by RIJNDAEL's s-box. Calling this new version KASUMI-R, it is found to have an FI function significantly better than others.

Finally, the randomness characteristics of the overall KASUMI-R for different rounds are compared to those of MISTY1 and KASUMI, in terms of avalanche weight distribution (AWD) and some statistical tests. The overall performance of the three ciphers is found to be same, although there is a significant difference in their FI functions.

Keywords: block cipher, KASUMI, MISTY1, SAC, linear approximation table, XOR table distribution, AWD.

ÖZ

MISTY1, KASUMI ve KASUMI-R'NİN ÇIĞ ÖZELLİKLERİ ÜZERİNE

AKLEYLEK, Sedat
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Melek Diker YÜCEL

Şubat 2008, 69 sayfa

Küresel taşınabilir iletişim sistemi(GSM) en yaygın olarak kullanılan cep telefonu teknolojisidir. GSM'de gizlilik 3. nesile kadar akan şifrelerin bazı uyarlamaları kullanılarak sağlanmıştır. Blok şifre olan KASUMI 3. nesilde kullanılmak üzere standart olarak seçilmiştir.

Bu tezde, KASUMI, MISTY1(KASUMI'nin önceki versiyonu) ve RIJNDAEL'un (gelişkin şifreleme standardı) değiştirme kutuları, doğrusallığa yakınsama tabloları, XOR tablo dağılımları ve katı çığ ölçütünü sağlayabilmeleri konularında değerlendirilmiştir. Daha sonra, KASUMI ve MISTY1'in doğrusal olmayan FI fonksiyonu katı çığ ölçütüne göre incelenmiştir. KASUMI'nin değiştirme kutuları RIJNDAEL'un değiştirme kutusu ile yer değiştirilerek yeni bir FI fonksiyonu elde edilmiştir. Bu yeni FI fonksiyonuna KASUMI-R adı verilip, performansının diğerlerinden oldukça iyi olduğu da gözlenmiştir.

Sonunda, çığ ağırlık dağılımı ve bazı sayımlamalı testler açısından KASUMI-R'nin tüm sisteminin farklı döngü sayıları için rasgelelik özelliklerinin MISTY1 ve KASUMI ile karşılaştırılması yapılmıştır. Şifrelerin FI fonksiyonları arasında dikkate değer fark bulunmasına rağmen, şifrelerin tüm performansları aynı bulunmuştur.

Anahtar Kelimeler : blok şifre, KASUMI, MISTY1, katı çığ ölçütü, doğrusallığa yakınsama tablosu, XOR tablo dağılımı, çığ ağırlık dağılımı.

To my family,

Acknowledgements

My supervisor, Assoc. Prof. Dr. Melek Diker Yücel, with her enthusiasm, her inspiration and her great efforts, became a model for me. It is difficult to overstate my gratitude to her. Throughout my thesis writing period, she provided encouragement, good teaching and lots of good ideas. This thesis would not have been completed without her guidance.

I am grateful to all my friends, who were with me, for their patience and understanding.

I also would like to thank to all people at the Institute of Applied Mathematics.

Lastly, and most importantly, I wish to thank my parents and my brothers.

TABLE OF CONTENTS

PLAGIARISM.. iv

ABSTRACT.. v

ÖZ.. vi

ACKNOWLEDGEMENTS...viii

LIST OF TABLES.. xii

LIST OF FIGURES.. xiv

LIST OF ABBREVIATIONS.. xv

CHAPTER

1 INTRODUCTION.. 1

1.1 Background.. 1

1.2 Scope and Objective of Thesis... 3

2 STRUCTURE OF MISTY1 AND KASUMI.............................. 5

2.1 MISTY1.. 5

2.1.1 FO Function of MISTY1.. 7

2.1.2 FL Function of MISTY1.. 10

2.1.3 Key Schedule of MISTY1.. 11

2.2 KASUMI.. 13

2.2.1 FO Function of KASUMI... 13

2.2.2 FL Function of KASUMI... 17

2.2.3 Key Schedule of KASUMI... 18

2.3 Descriptions of 7x7 and 9x9 S-boxes.. 19

2.3.1 MISTY1’s and KASUMI’s 7x7 S-boxes........................ 19

2.3.2 MISTY1’s and KASUMI’s 7x7 S-boxes........................ 20

3 TEST CRITERIA FOR BOOLEAN FUNCTIONS................... 22

3.1 Boolean Functions.. 22
3.2 Cryptographic Properties of Boolean Functions........................... 24
3.2.1 Completeness and Avalanche Criterion......................... 24
3.2.2 Strict Avalanche Criterion.. 25
3.2.3 Nonlinearity... 26
3.2.4 Linear Approximation Table (LAT) Distribution........... 27
3.2.5 Exclusive OR-XOR Table Distribution......................... 28
4 S-BOX TEST RESULTS FOR MISTY1, KASUMI AND RIJNDAEL... 30
4.1 Test Results of LAT Distribution.. 30
4.1.1 Results for the 7x7 S-boxes of MISTY1 and KASUMI. 31
4.1.2 Results for the 9x9 S-boxes of MISTY1 and KASUMI. 32
4.1.3 Results for the 8x8 S-box of RIJNDAEL....................... 33
4.1.4 Comparison of MISTY1, KASUMI and RIJNDAEL S-boxes... 34
4.2 Test Results of XOR Table Distribution...................................... 36
4.2.1 Results for the 7x7 S-boxes of MISTY1 and KASUMI. 36
4.2.2 Results for the 9x9 S-boxes of MISTY1 and KASUMI. 37
4.2.3 Results for the 8x8 S-box of RIJNDAEL....................... 39
4.2.4 Comparison of MISTY1, KASUMI and RIJNDAEL S-boxes... 39
4.3 Test Results of Strict Avalanche Criterion for S-boxes............... 40
4.3.1 Results for the 7x7 S-boxes of MISTY1 and KASUMI. 41
4.3.2 Results for the 9x9 S-boxes of MISTY1 and KASUMI. 42
4.3.3 Results for the 8x8 S-box of RIJNDAEL....................... 43
4.3.4 Comparison of MISTY1, KASUMI and RIJNDAEL S-boxes... 44
4.4 Test Results of Strict Avalanche Criterion for FI Functions......... 45
4.4.1 FI Function with RIJNDAEL S-box.............................. 45
4.4.2 Results for the FI Function of MISTY1......................... 47
4.4.3 Results for the FI Function of KASUMI....................... 47

4.4.4 Results for the FI Function of KASUMI with RIJNDAEL S-box... 48

4.4.5 Comparison of MISTY1, KASUMI and KASUMI with RIJNDAEL S-boxes FI Functions.. 49

5 RANDOMNESS CRITERIA FOR BLOCK CIPHERS............. 53

5.1 Avalanche Weight Distribution.. 53

5.1.1 AWD Test Results for MISTY1.................................... 54

5.1.2 AWD Test Results for KASUMI................................... 55

5.1.3 AWD Test Results for KASUMI-R................................ 56

5.1.4 Comparison of MISTY1, KASUMI and KASUMI-R in view of AWD.. 57

5.2 Randomness Testing... 58

5.2.1 Frequency (Monobit) Test.. 58

5.2.1.1 Test Description... 58

5.2.2 Frequency Test within a Block....................................... 59

5.2.2.1 Test Description... 59

5.3 Test Results for Randomness.. 60

5.3.1 Frequency within a Block Test Results for MISTY1, KASUMI and KASUMI-R.. 62

5.4 Comparison of Avalanche Weight Distribution and Randomness Test Results... 65

6 CONCLUSION... 66

REFERENCES... 68

LIST OF TABLES

TABLES

2.1 Round Subkeys of MISTY1 12

2.2 Round Subkeys of KASUMI 18

2.3 Constants for Key Schedule of KASUMI 18

4.1 LAT Distribution of MISTY1’s 7x7 S-box for Single bit Input and Output Differences 31

4.2 LAT Distribution of KASUMI’s 7x7 S-box for Single bit Input and Output Differences 32

4.3 LAT Distribution of MISTY1’s 9x9 S-box for Single bit Input and Output Differences 33

4.4 LAT Distribution of KASUMI’s 9x9 S-box for Single bit Input and Output Differences 33

4.5 LAT Distribution of RIJNDAEL’s 8x8 S-box for Single bit Input and Output Differences 34

4.6 Distribution Percentages of LAT Elements for the Three S-boxes 35

4.7 Probability Biases for the Three S-boxes 35

4.8 Nonlinearity for the Three S-boxes 36

4.9 XOR Table Distribution of MISTY1’s 7x7 S-box for Single bit Input and Output Differences 37

4.10 XOR Table Distribution of KASUMI’s 7x7 S-box for Single bit Input and Output Differences 37

4.11 XOR Table Distribution of MISTY1’s 9x9 S-box for Single bit Input and Output Differences 38

4.12 XOR Table Distribution of KASUMI’ 9x9 S-box for Single bit Input and Output Differences 38

4.13 XOR Table Distribution of RIJNDAEL’s 8x8 S-box for Single bit Input and Output Differences 39

4.14 Differential Uniformity of XOR Table Elements for Each Row of the Three S-boxes 40

4.15 XOR Table Elements for Each Row of the Three S-boxes 40

4.16 Normalized Distance to SAC for S-boxes 44

4.17 Normalized Distance to SAC for FI functions 52

4.18 Probabilities for FI Function in terms of SAC 52

5.1 Resemblance Percentages $R_r^{e_1}$ for MISTY1, KASUMI and KASUMI-R 58

6.1 Summary of Test Results for S-boxes 66

LIST OF FIGURES

FIGURES

2.1 Two Rounds of MISTY1 Encryption Process 6

2.2 FO Function of MISTY1 8

2.3 FI Function of MISTY1 9

2.4 FL Function of MISTY1 11

2.5 Key Schedule of MISTY1 12

2.6 Two Rounds of KASUMI Encryption Process 14

2.7 FO Function of KASUMI 15

2.8 FI Function of KASUMI 16

2.9 FL Function of KASUMI 17

4.1 Number of Occurrences of LAT Elements for RIJNDAEL's S-box 34

4.2 Normalized Distance to SAC of MISTY1 7 x 7 S-box 41

4.3 Normalized Distance to SAC of KASUMI 7 x 7 S-box 42

4.4 Normalized Distance to SAC of MISTY1 9 x 9 S-box 43

4.5 Normalized Distance to SAC of KASUMI 9 x 9 S-box 43

4.6 Normalized Distance to SAC of RIJNDAEL 8 x 8 S-box 44

4.7 FI Function of KASUMI-R 46

4.8 Normalized Distance to SAC of MISTY1's FI Function 47

4.9 Normalized Distance to SAC of KASUMI's FI Function 48

4.10 Normalized Distance to SAC of KASUMI-R's FI Function 49

4.11 Normalized Distance to SAC of FI Functions wıth All Values 50

4.12 Normalized Distance to SAC of FI Functions 51

5.1 Ideal AWD Curve 53

5.2 AWD Curves of MISTY1 for e_1 for all Rounds 55

5.3 AWD Curves of KASUMI for e_1 for all Rounds 56

5.4 AWD Curves of KASUMI-R for e_1 for all Rounds 56

5.5 Percentages of 1's for Avalanche Plaintext 63

5.6 Percentages of 1's for Plaintext/Ciphertext Correlation 64

5.7 Percentages of 1's for Low Density Plaintext 65

LIST OF ABBREVIATIONS

1G	First Generation
2G	Second Generation
2.5G	Second and a half Generation
3G	Third Generation
3GPP	3rd Generation Partnership Project
AES	Advanced Encryption Standard
AWD	Avalanche Weight Distribution
CEPT	The Conference of European Posts and Telegraphs
CRYPTREC	Cryptography Research and Evaluation Committee
ETSI	European Telecommunication Standards Institute
GPRS	General Packet Radio Service
GSM	Global System for Mobile
LAT	Linear Approximation Table
LFSR	Linear Feedback Shift Register
NIST	The US National Institute of Standards and Technology
SAC	Strict Avalanche Criterion
UMTS	Universal Mobile Telecommunications System
XOR	Exclusive or
XOR Table	Differential Table Distribution
WAP	Wireless Application Protocol

CHAPTER 1

INTRODUCTION

1.1 Background

Cryptographic techniques have been used for many centuries to protect the secrecy of diplomatic correspondence and military communications. Today, developments on computer and communication sciences have helped transferring a big amount of data through the long distance channels. These data must be protected in several ways to provide confidentiality, integrity and authentication. Cryptology is the science, which answers all such needs in today's communication systems.

Cryptology is used to provide security in public applications such as e-government applications, electronic commerce, credit cards, wireless connections, GSM mobile phones. In the Global System for Mobile (GSM) communications the standard for mobile phones, the mobile system used all over the world, there is a built-in cipher to ensure that your conversation is private. The cryptographic algorithms of GSM have received a lot of interest and activity from the cryptographic research community.

GSM has some different generations in use such as first generation (1G), and second generation (2G). In the beginning of 1980s, analog cellular telephone systems (1G, First Generation) were started to be used with rapid growth in Europe. The Conference of European Posts and Telegraphs (CEPT) formed a study group called the Groupe Spécial Mobile (GSM) in 1982 to create a digital standard (2G, Second Generation) for European mobile system [16]. In 1989, GSM responsibility was transferred to the European Telecommunication Standards Institute (ETSI), and they published the first GSM specifications in 1990, where the meaning of GSM was changed to Global System for Mobile [10].

Since the nature of the wireless communication is more susceptible to attacks than wired communication, to provide security is very important for GSM. In the second generation, cryptographic techniques available for GSM were A5/1 and A5/2, both of which use stream ciphers. A5/1 was the first cryptographic GSM algorithm developed to provide privacy in 1987. The A5/1 is based on the output of three LFSRs (Linear Feedback Shift Register). It was demonstrated in [3] that A5/1 could be cracked in less than 1 second on a desktop PC. The attack uses statistical analysis and exploits its poor avalanche properties. In 1989 A5/2 was developed to overcome export problem of A5/1 [21]. A5/2 containing four LFSRs was used to provide voice privacy for a short time as it was cryptanalyzed in the same month that it was published [27]. A5/2 is a weakened version of A5/1. The approximate design of A5/1 was leaked in 1994 and the exact design of both A5/1 and A5/2 was reverse engineered from an actual GSM telephone in 1999 [4]**.** It is interesting that only after their design became publicly known; they were cracked by using reverse engineering.

With the technological developments, by the end of 1990's GSM could handle different types of services: high quality encrypted voice transmissions, short message servicing, fax services, Wireless Application Protocol (WAP) promoted Internet applications [10]. Later, a group of studies called second and a half generation (2.5G) provided mobile Internet supporting services (Internet browsing, e-mail and multimedia messages such as the general packet radio service-GPRS which enables larger packets of data to be sent). Hence, 2.5G became a step between 2G and 3G.

Universal Mobile Telecommunications System (UMTS), 3G, is a further development of the 2.5G. In order to make this communication secure, a new security algorithm was chosen in 2002 by 3rd Generation Partnership Project (3GPP) founded with the duty of defining world wide trusted standards for 3G [22].

The next generation of A5 algorithm in contrast to previous ones has been made available to the public. In 2002, A5/3 was added to GSM encryption algorithms. A5/3 is based on the block cipher KASUMI declared as the standard cryptographic algorithm for UMTS applications [22]. KASUMI was specially developed to gain public confidence in UMTS security.

1.2 Scope and Objective of Thesis

This thesis is intended to analyze the cryptographic strength of the block cipher KASUMI using some cryptographic test criteria. Since KASUMI is a variant of MISTY1 recommended for Japanese government use by the Cryptography Research and Evaluation Committee (CRYPTREC project) [23] in 2003, MISTY1 is investigated, too.

Rijndael's s-box is only used as a reference for comparison with the s-boxes of MISTY1 and KASUMI; since Rijndael, selected as Advanced Encryption Standard (AES) [5] by the US National Institute of Standards and Technology (NIST) in 2000, appears to have an adequate security margin and its security is approved by the society of cryptographers.

MISTY1's, KASUMI's and RIJNDAEL's s-boxes are investigated in terms of some cryptographic test criteria for Boolean functions appearing in the literature such as completeness, avalanche, strict avalanche, nonlinearity, linear approximation table (LAT) and differential table distribution (XOR). S-boxes of the FI function, which is the core of KASUMI, are then replaced with the s-box of RIJNDAEL. We call this new cipher KASUMI-R. Then, FI functions of all the three ciphers are analyzed according to the strict avalanche criterion. In addition, the overall performance of MISTY1, KASUMI and KASUMI-R is measured by the avalanche weight distribution and some of the statistical tests for randomness, based on NIST Statistical Test Suite [15].

In Chapter 2, the structures of the block ciphers MISTY1, KASUMI are considered in detail. The differences between MISTY1 and KASUMI, and some observations about their components are discussed. Moreover, descriptions of the s-boxes are given in this chapter.

In Chapter 3, the theory of some well known cryptographic test criteria for Boolean functions appearing in the literature; such as completeness, avalanche, strict avalanche, nonlinearity, LAT and XOR table distribution are described.

In Chapter 4, the s-boxes of MISTY1, KASUMI and RIJNDAEL are investigated according to the test criterion defined in Chapter 3. Then, we define a new FI function for KASUMI-R. The FI functions of MISTY1, KASUMI and KASUMI-R are analyzed according to the strict avalanche criterion. The test results are presented and discussed.

In Chapter 5, avalanche weight distribution and two core tests of the NIST Statistical Test Suite, monobit test and frequency test within a block, are defined to examine the cipher with all components and see its randomness properties. The test results are presented and compared.

Chapter 6 summarizes the results of the work done in this thesis.

CHAPTER 2

STRUCTURES OF MISTY1 AND KASUMI

MISTY1 [13] is an encryption algorithm developed by Mitsubishi Electric and submitted to New European Schemes for Signature, Integrity and Encryption (NESSIE) project [25]. MISTY1's security capabilities are later used as the base for KASUMI [22], which has become the international encryption standard for the 3rd generation mobile phones.

This chapter gives detailed descriptions of MISTY1 in Section 2.1, and followed by KASUMI in Section 2.2. Finally, the 7x7 and 9x9 s-boxes of MISTY1 and KASUMI are compared in Section 2.3.

2.1 MISTY1

MISTY1, recommended for Japanese government use by the Cryptography Research and Evaluation Committee (CRYPTREC project) [23] in 2003, was first published in 1996. It uses Feistel structure, which takes a 64-bit plaintext and a 128-bit key to produce a 64-bit output. It is recommended for a multiple of 4 rounds typically as 8 rounds. The entire algorithm is built from recursive small components. This recursive design adds a lot of complexity to the cipher, making its analysis harder. MISTY1 is the first block cipher designed for practical use with provable security against differential and linear cryptanalysis [12].

Let E be the encryption function of MISTY1 $E:\{0,1\}^{64}\,\mathrm{x}\,\{0,1\}^{128} \rightarrow \{0,1\}^{64}$ that takes the two inputs a 64-bit plaintext P and a 128-bit key K, to return a 64-bit ciphertext $C:E(P,K)$.

$E(P,K)$ decomposes into subfunctions FL and FO, both of which operating on half of the input text. $FL(p,KL)$ is a linear function which maps a 32-bit block p to a 32-bit sequence, using a 32-bit key KL. The other subfunction $FO(p, KO, KI)$ is not linear but also maps a 32-bit block *p* to a 32-bit block, using a 64-bit key KO and a 48-bit key KI.

Encryption process is summarized as follows:

$$P = L_0 \parallel R_0 \qquad (2.1.1)$$

For $i = 1,3,5,7$

$$\begin{aligned} R_i &= FL(L_{i-1}, KL_i) \\ L_i &= FL(R_{i-1}, KL_{i+1}) \oplus FO(R_i, KO_i, KI_i) \\ T &= L_i \qquad (2.1.2) \\ L_i &= R_i \oplus FO(L_i, KO_{i+1}, KI_{i+1}) \\ R_i &= T \end{aligned}$$

For $i = 2,4,6,8$

$$\begin{aligned} R_i &= FO(L_{i-1}, KO_i, KI_i) \\ L_i &= R_i \oplus R_{i-1} \end{aligned} \qquad (2.1.3)$$

where the output is $C = L_8 \parallel R_8$ (2.1.4), and P: 64-bit, L_i: 32-bit, R_i: 32-bit, KL_i: 32-bit, KI_i: 48-bit, KO_i: 64-bit, C: 64-bit.

Encryption process for two rounds can be seen in Figure 2.1.

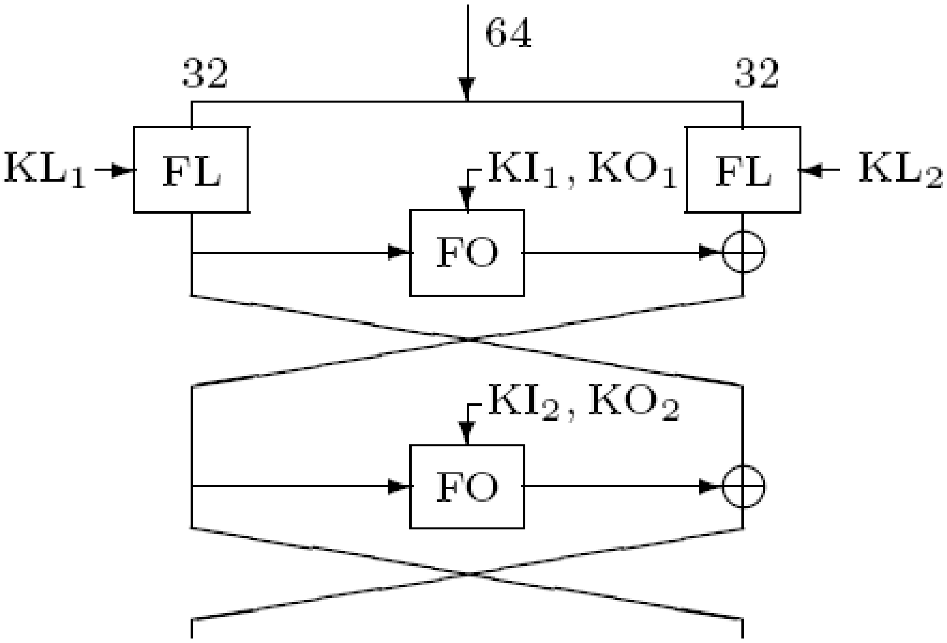

Figure 2.1 Two Rounds of MISTY1 Encryption Process

The main component of the $FO(p,KO,KI)$ is called the FI function, which maps 16 input bits to 16 output bits. $FI(q,KI)$ function, where q is the 16-bit input and KI is the 48-bit key, uses two s-boxes, a 7x7 s-box, S_7, and a 9x9 s-box, S_9. The keys KL, KO and KI are produced from the initial key K as described in Section 2.1.3.

2.1.1 FO Function of MISTY1

The $FO(p,KO,KI)$ function, which is the data randomizing part of MISTY1, maps a 32-bit input *p* to a 32-bit output. The function uses two subkeys, a 64-bit KO_i and a 48-bit KI_i. The main part of the FO function shown in Figure 2.2 is the $FI(q,KI)$ function, where *q* is a 16-bit word. Since each branch of the $FO(p,KO,KI)$ function works on 16-bit words, $p = L_0 \| R_0$, $KO_i = KO_{i1} \| KO_{i2} \| KO_{i3} \| KO_{i4}$ and $KI_i = KI_{i1} \| KI_{i2} \| KI_{i3}$ are all divided into 16-bit words. Then, $FO(p,KO,KI)$ is defined as follows:

For $j = 1$ to 3 do

$$\begin{aligned} R_j &= FI\left(L_{j-1} \oplus KO_{ij}, KI_{ij}\right) \oplus R_{j-1} \\ L_j &= R_{j-1} \end{aligned} \qquad (2.1.5)$$

$$C = \left(L_3 \oplus KO_{i4}\right) \| R_3 \qquad (2.1.6)$$

where C is the output.

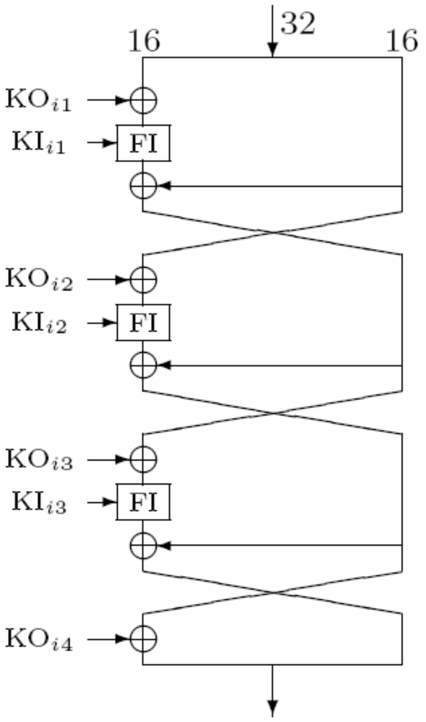

Figure 2.2 FO function of MISTY1

The FI function is the core of the FO function. It maps 16 input bits to 16 output bits and uses 7x7 s-box, S_7, and 9x9 s-box, S_9. S-boxes are incorporated into the lowest level of a recursively constructed Feistel structure. They are designed to obtain good resistance to linear and differential attacks. S_7 comprises a set of cubic functions, S_9 comprises a set of quadratic functions. In selecting S_7 and S_9, designers of MISTY1 have used the following criteria:

1. Their average linear/differential probability must be minimal.
2. Their algebraic degree should be as high as possible.

Details of descriptions of the s-boxes S_7 and S_9 are given in Section 2.3. The FI function also uses two additional functions, which are designated by the ZE (i.e., zero extend) function that appends two zeros before the most significant bit of a 7-bit string and the TR (i.e., truncate) function that discards two most significant bit of a 9-bit string.

The 16-bit input of the $FI(q,KI)$ function is split into two unequal components, a 9-bit left half and a 7-bit right half, where $q = L_0 \parallel R_0$. Similarly, the subkey KI_{ij} is split into a 7-bit component KI_{ij1} and a 9-bit component KI_{ij2}, where $KI_{ij} = KI_{ij1} \| KI_{ij2}$. The $FI(q, KI)$ function is then defined as follows:

$$\begin{aligned} R_1 &= S9[L_0] \oplus ZE[R_0] \\ L_1 &= R_0 \\ R_2 &= S7[L_1] \oplus TR[R_1] \oplus KI_{ij1} \\ L_2 &= R_1 \oplus KI_{ij2} \\ R_3 &= S9[L_2] \oplus ZE[R_2] \\ L_3 &= R_2 \end{aligned} \tag{2.1.7}$$

$$C = L_3 \| R_3 \tag{2.1.8}$$

where C is the output and $S7$ and $S9$ are the s-box functions. The FI function is depicted in Figure 2.3.

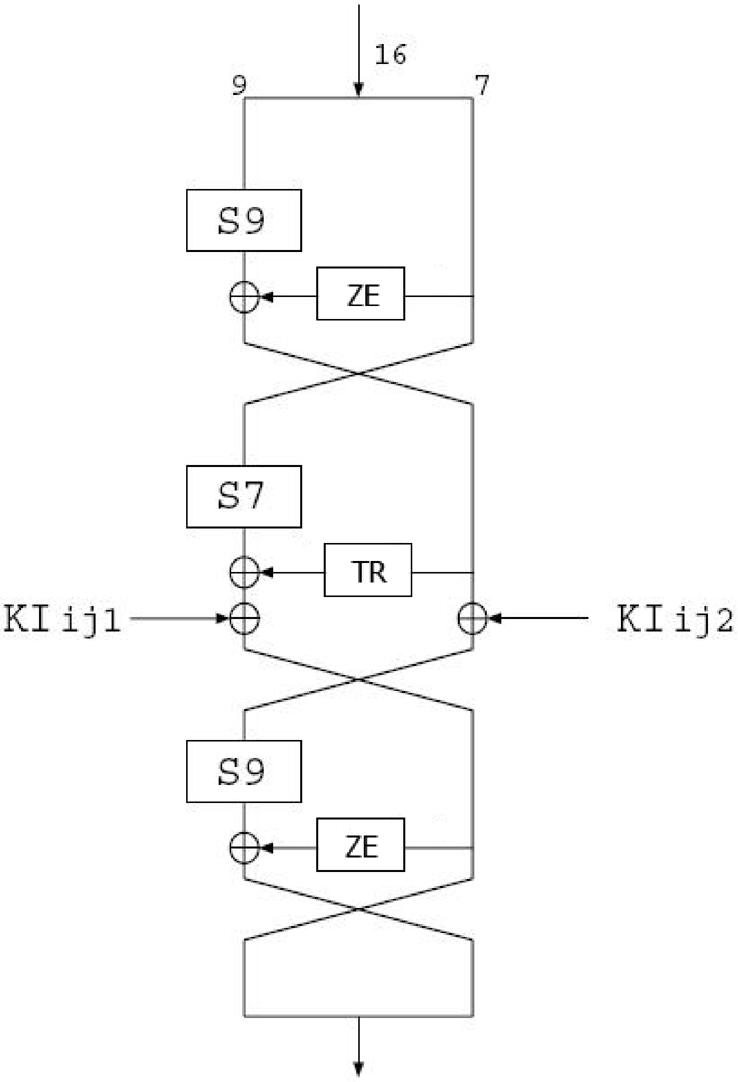

Figure 2.3 FI Function of MISTY1

2.1.2 FL Function of MISTY1

The $FL(p, KL)$ function is a linear function used for the diffusion, i.e., it makes individual bits harder to follow through the rounds. Since this function is linear as long as the key is fixed, it does not affect the average linear/differential probability of the entire algorithm.

The $FL(p, KL)$ function receives a 32-bit input p and a 32-bit subkey KL_i. It gives a 32-bit output C.

The input is split this into two 16-bit halves, where $p = L_0 \| R_0$. Similarly, subkey KL_i is divided two 16-bit halves, where $KL_i = KL_{i1} \| KL_{i2}$. Then, the FL function is defined as follows :

$$\begin{aligned} C_R &= (L_0 \cap KL_{i1}) \oplus R_0 \\ C_L &= (C_R \cup KL_{i2}) \oplus L_0 \end{aligned} \qquad (2.1.9)$$

$$C = C_L \| C_R \qquad (2.1.10)$$

where $\cap$ and $\cup$ are the logical AND and OR operations, respectively. The FL function is shown in Figure 2.4.

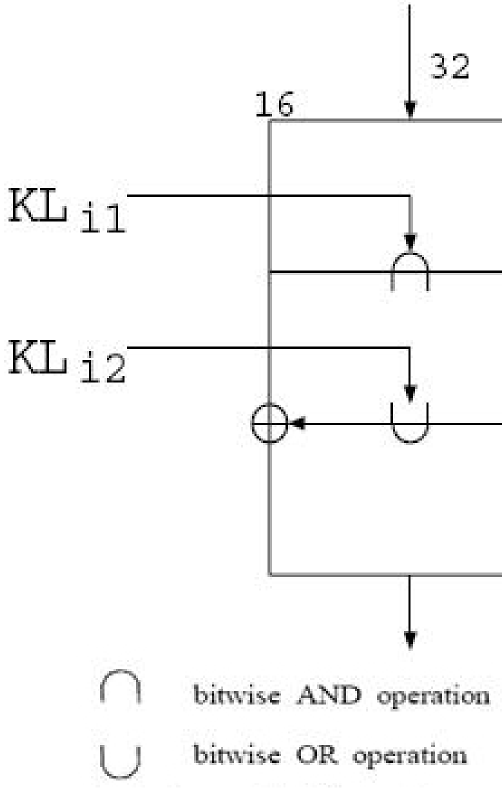

Figure 2.4 FL Function of MISTY1

2.1.3 Key Schedule of MISTY1

Key schedule comprises 8 consecutive applications of the FI function. Firstly, the 128-bit key, K, is split into eight parts, $K = K_1 \| K_2 \| ... \| K_8$, each of length 16-bit. Then, K_i for $1 \le i \le 8$ is considered as the input to FI (see Figure 2.3) with $K_{i+1(\mathrm{mod}\,8)}$ acting as the key to the FI function, the 16-bit output from each FI function is extra keys, K_i', $1 \le i \le 8$. These are used in the FI and FL functions. Key schedule, which uses FI function, is shown in Figure 2.5.

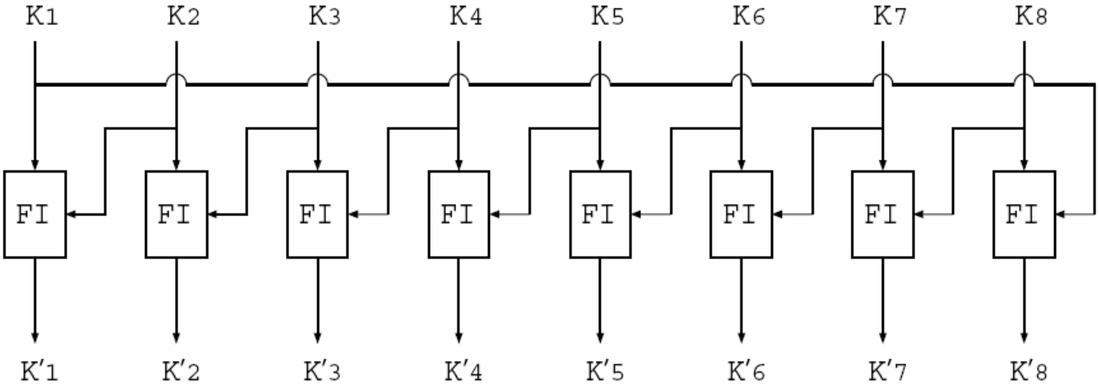

Figure 2.5 Key Schedule of MISTY1

Round subkeys, KL, KO, KI, are demonstrated in Table 2.1 for the i^{th} round. First two rows show 32 bits of KL, the next four and the last three rows denote the 64 and 48 bits of KO and KI, respectively. Note that all KI's are the output of the FI function.

Table 2.1 Round Subkeys of MISTY1

Subkeys	i^{th} **Round Output**
KL_{i1}	$K_{\frac{i+1}{2}}$ (odd i) $K'_{\frac{i}{2}+2}$ (even i)
KL_{i2}	$K'_{\frac{i+1}{2}+6(\bmod 8)}$ (odd i) $K_{\frac{i}{2}+4}$ (even i)
KO_{i1}	K_i
KO_{i2}	$K_{i+2(\bmod 8)}$
KO_{i3}	$K_{i+7(\bmod 8)}$
KO_{i4}	$K_{i+4(\bmod 8)}$
KI_{i1}	$K'_{i+5(\bmod 8)}$
KI_{i2}	$K'_{i+1(\bmod 8)}$
KI_{i3}	$K'_{i+3(\bmod 8)}$

2.2 KASUMI

Within the security architecture of the Third Generation Partnership Project (3GPP) system there are two standardized algorithms: A confidentiality algorithm *f8* and an integrity algorithm *f9* [24]. Each of these algorithms is based on the KASUMI algorithm. KASUMI, eight round Feistel network, is a block cipher that produces a 64-bit output from a 64-bit input under the control of a 128-bit key. The differences between MISTY1 and KASUMI are also emphasized below while explaining KASUMI's structure.

2.2.1 FO Function of KASUMI

The basic structure of KASUMI is very similar to MISTY1. KASUMI also consists of the subfunctions FL, FO and FI that are used in conjunction with associated subkeys KL, KO and KI. The overall structure of KASUMI is a 64-bit permutation composed of eight rounds of Feistel network. The round function consists of a non-linear mixing function FO and linear mixing function FL.

Let $E(P,K)$ be the encryption function of KASUMI $E:\{0,1\}^{64}\text{x}\{0,1\}^{128}\rightarrow\{0,1\}^{64}$ that returns a 64-bit output $C:E(P,K)$. The encryption function (see Figure 2.6) is slightly different from MISTY1's and summarized as follows:

Encryption process of KASUMI can be summarized as follows:

$$P = L_0 \,\|\, R_0 \qquad (2.2.1)$$

For $i = 1,3,5,7$

$$\begin{aligned} R_i &= L_{i-1} \\ L_i &= FO\big(FL(R_{i-1}, KL_i), KO_i, KI_i\big) \\ T &= L_i \qquad (2.2.2) \\ L_i &= R_i \oplus T \\ R_i &= T \end{aligned}$$

For $i = 2,4,6,8$

$$\begin{aligned}
R_i &= L_{i-1} \\
L_i &= FL(FO(R_i, KO_i, KI_i), KL_i) \\
T &= L_i \\
L_i &= R_i \oplus T \\
R_i &= T
\end{aligned} \tag{2.2.3}$$

where the output is $C = L_8 \| R_8$ (2.2.4) and P: 64-bit, L_i: 32-bit, R_i: 32-bit, KL_i: 32-bit, KI_i: 48-bit, KO_i: 48-bit, C: 64-bit.

In KASUMI (see Figure 2.6), the FL function precedes FO function in the odd rounds and it follows the FO function in the even rounds. On the other hand, in MISTY1, the FL function is used in both branches of the odd rounds; however, it is not used at all in the even rounds (see Figure 2.1).

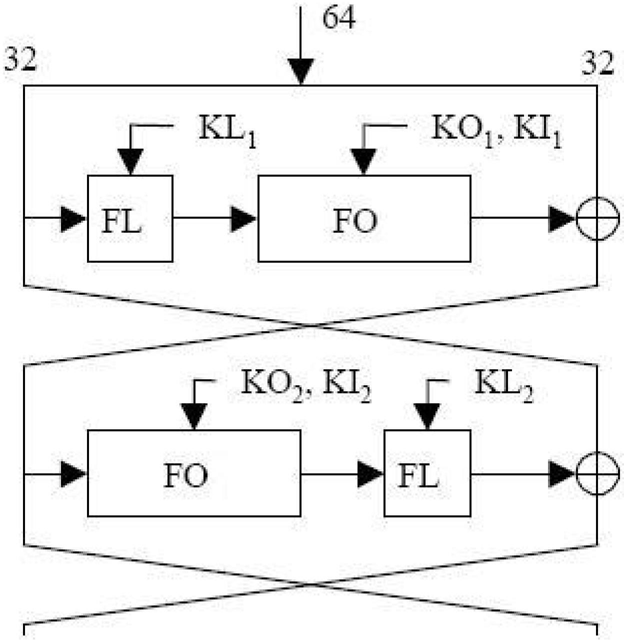

Figure 2.6 Two Rounds of KASUMI Encryption Process

The $FO(p, KO, KI)$ function of the KASUMI is almost the same as the $FO(p, KO, KI)$ of MISTY1 as described in Section 2.1.1, except for the missing last step i.e., Eq. 2.1.6. For this reason, the required key length for KI is 48 bits for KASUMI whereas it is 64 bits for MISTY1.

The 48-bit subkeys, KO_i and KI_i are subdivided into three 16-bit subkeys, where $KO_i = KO_{i1} \| KO_{i2} \| KO_{i3}$ and $KI_i = KI_{i1} \| KI_{i2} \| KI_{i3}$. Then, the FO function is defined as follows :

For $j = 1$ to 3 do

$$\begin{aligned} R_j &= FI\left(L_{j-1} \oplus KO_{ij}, KI_{ij}\right) \oplus R_{j-1} \\ L_j &= R_{j-1} \end{aligned} \tag{2.2.5}$$

$$C = L_3 \| R_3 \tag{2.2.6}$$

where C is the output.

Moreover, the $FI(q, KI)$ function (see Figure 2.8) uses slightly different s-boxes, which are also 7x7 and 9x9 called S_7 and S_9 (see Section 2.3).

The FO function of KASUMI is shown in Figure 2.7.

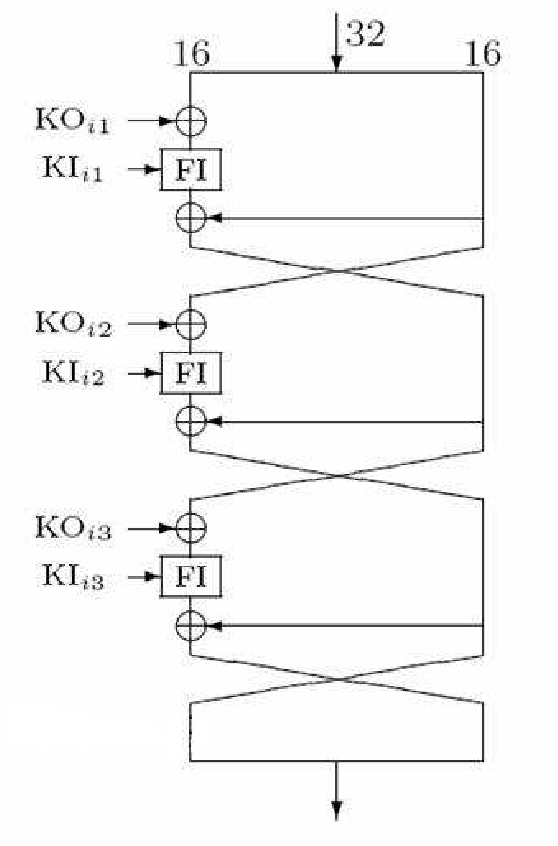

Figure 2.7 FO function of KASUMI

The second difference of KASUMI's FI function is its additional round (see 4th round in Figure 2.8) as compared to MISTY1's (see Figure 2.3).

Using the definitions given for MISTY1, the FI function of KASUMI is defined as follows :

$$\begin{aligned}
R_1 &= S9[L_0] \oplus ZE[R_0] \\
L_1 &= R_0 \\
R_2 &= S7[L_1] \oplus TR[R_1] \oplus KI_{ij1} \\
L_2 &= R_1 \oplus KI_{ij2} \\
R_3 &= S9[L_2] \oplus ZE[R_2] \\
L_3 &= R_2 \\
R_4 &= R_3 \\
L_4 &= S7[L_3] \oplus TR[R_3]
\end{aligned} \tag{2.2.7}$$

$$C = L_4 \parallel R_4 \tag{2.2.8}$$

where C is the output.

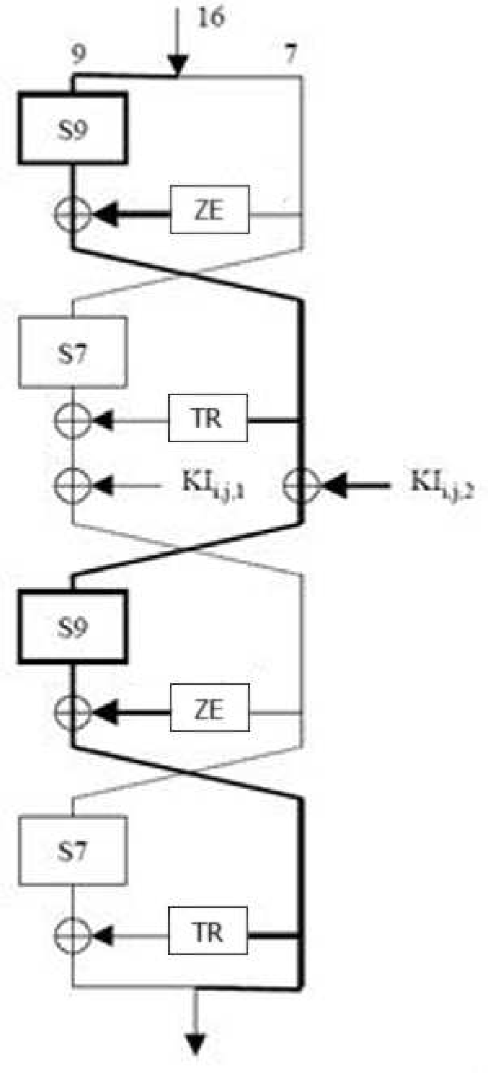

Figure 2.8 FI function of KASUMI

2.2.2 FL Function of KASUMI

FL function is quite similar to the MISTY1's. The only difference (see Figure 2.7) is the rotation operation ROT, which rotates its input one bit to left. Using the previously defined input and key vectors, the $FL(p, KL)$ function is defined as follows:

$$\begin{aligned} C_R &= ROT(L_0 \cap KL_{i1}) \oplus R_0 \\ C_L &= ROT(C_R \cup KL_{i2}) \oplus L_0 \end{aligned} \qquad (2.2.9)$$

$$C = C_L \parallel C_R \qquad (2.3.10)$$

where the output is C and, $\cap$ and $\cup$ are the logical AND and OR operations, respectively.

FL function in both MISTY1 and KASUMI has the property that for any subkey KL_i, an input of $P = L_0 \parallel R_0$. $L_0 = (00...0)_{1x16}$, $R_0 = (11...1)_{1x16}$ always give an output of all 1's. Hence for some round inputs, some of the key bits in KL_i can be changed without having any effect on the output of that round. This can be used to guarantee a zero difference at the end of the first round. Small changes to the input to FL function only make small output changes.

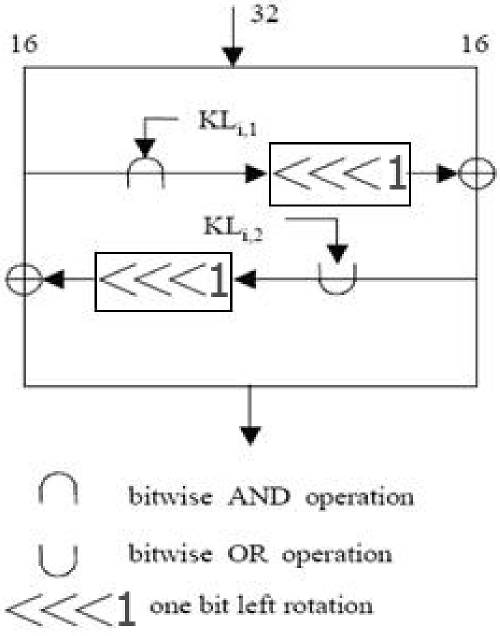

Figure 2.9 FL function of KASUMI

2.2.3 Key Schedule of KASUMI

The subkeys KL, KO and KI of KASUMI are produced using the original 128-bit key K, as shown in Table 2.2. Each column of Table 2.2 indicates the keys used for i^{th} round. First two rows denote the 32 bits of KL, the next and the last three rows show the 48 bits of KO and KI, respectively. $<<< l$ is equal to the rotation operation, which rotates its input l-bit to left.

The 128-bit key K is subdivided into eight 16-bit values $K = K_1 \| K_2 \| ... \| K_8$. A second array of subkeys, K' is derived from K by applying for $1 \le j \le 8$, $K'_j = K_j \oplus C_j$, where C_j is the constant value defined in Table 2.3 in hexadecimal form.

Table 2.2. Round Subkeys of KASUMI

Subkeys	i^{th} **Round Output**
KL_{i1} KL_{i2}	$K_i <<< 1$ $K'_{i+2(\mathrm{mod}\,8)}$
KO_{i1} KO_{i2} KO_{i3}	$K_{i+1(\mathrm{mod}\,8)} <<< 5$ $K_{i+5(\mathrm{mod}\,8)} <<< 8$ $K_{i+6(\mathrm{mod}\,8)} <<< 13$
KI_{i1} KI_{i2} KI_{i3}	$K'_{i+4(\mathrm{mod}\,8)}$ $K'_{i+3(\mathrm{mod}\,8)}$ $K'_{i+7(\mathrm{mod}\,8)}$

Table 2.3. Constants for Key Schedule of KASUMI

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
0x0123	0x4567	0x89AB	0xCDEF	0xFEDC	0xBA98	0x7654	0x3210

The 128 bit additional bits are computed by using a linear relation, but the outcome of 256 bit is used a little bit differently, as the bits chosen as round subkeys are rotated by several constants in some cases.

In MISTY1 the 128-bit key is used to compute 128-bit additional key using FI function. The outcome of 256-bit is used again and again in various orders as the round subkeys.

2.3 Descriptions of 7x7 and 9x9 S-boxes

The s-boxes S_7 and S_9 are obtained as affine transforms of power functions over the corresponding fields, with Kasami's and Gold's exponents. S_7 and S_9 are designed with function $x \to x^{81}$ in F_2^7 and $x \to x^5$ in F_2^9, respectively.

Firstly, the 81th and 5th power of the element in F_2^7 and F_2^9 is found for every nonzero element, respectively. Then, the result is transformed by an affine transformation to produce the output. The effect of the affine transform is to remove fixed points.

2.3.1 MISTY1's and KASUMI's 7x7 S-boxes

Definition 2.1 : Kasami's exponents [6] implies that for $n = 2m+1,\ 2 \le k \le m$ and $\gcd(k,n) = 1$, the power function x^d, where $d = (2^{2k} - 2^k + 1)\bmod(2^n - 1)$ is almost perfect nonlinear. Moreover, exponent d' is equivalent to d if there is an integer t such $d' = 2^t d$ i.e., the power functions, x^d and $x^{d'}$ have the same properties.

For $n = 7$, $d = 2^4 - 2^2 + 1 = 13$ is a Kasami exponent with $k = 2$. Then, with $d' = 2^4 d$, one can design S_7 with $81 = 2^4(2^4 - 2^2 + 1)\bmod(2^7 - 1)$. The Hamming weight of d, the degree of S_7, is 3 since $13 = 2^4 - 2^2 + 2^0$. S_7 is constructed by composing two transformations :

i) Take the 81th power of the element in F_2^7.

ii) Apply the affine transformation :

$y_i = x_i \oplus x_{(i+3)\bmod 7} \oplus x_{(i+4)\bmod 7} \oplus x_{(i+6)\bmod 7} \oplus c_i$, for $0 \le i < 7$, where x_i is the 81th power of the element in F_2^7 and c is with value (0011011) for MISTY1.

$y_i = x_{(i+6)\bmod 7} \oplus c_i$, for $0 \le i < 7$, where x_i is the 81th power of the element in F_2^7 and c is with value (0110110) for KASUMI.

Calling an element $x \in F_2^7$, $x = (x_0 ... x_6)$ the s-box output is $y = A \cdot x + c$ with $c = (c_0 ... c_6)$ if x is equal to the 81th power of the s-box input.

An affine transformation can be expressed in the matrix form for MISTY1 in (2.3.1) and for KASUMI in (2.3.2) as :

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad (2.3.1)$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (2.3.2)$$

2.3.2 MISTY1's and KASUMI's 9x9 S-boxes

Definition 2.2 : Gold's exponents [6] implies that for $n = 2m+1,\ 1 \le k \le m$ and $\gcd(k, n) = 1$, the power function x^d, where $d = (2^k + 1)\bmod(2^n - 1)$ is almost perfect nonlinear.

Gold exponent is calculated to design S_9 such that $516 = 2^9 + 2^2 \bmod(2^9 - 1) = 2^2 + 1$, since for $n = 9$, $d = 2^2 + 1 = 5$ is a Gold exponent with $k = 2$. S_9 is constructed by composing two transformations :

i) Take the 5th power of the element in F_2^9.

ii) Apply the affine transformation :

$y_i = x_{(i+2)\bmod 9} \oplus x_{(i+4)\bmod 9} \oplus x_{(i+6)\bmod 9} \oplus x_{(i+8)\bmod 9} \oplus c_i$, for $0 \le i < 9$, where x_i is the 5th power of the element in F_2^9 and c is with value (110000111) for MISTY1.

$y_i = x_{(i+3)\bmod 9} \oplus x_{(i+6)\bmod 9} \oplus c_i$, for $0 \le i < 9$, where x_i is the 5th power of the element in F_2^9 and c is with value (111001010) for KASUMI.

Calling an element $x \in F_2^9$, $x = (x_0 ... x_8)$ the s-box output is $y = A \cdot x + c$ with $c = (c_0 ... c_8)$ if x is equal to the 5th power of the s-box input.

Matrix form of the 9x9 s-boxes of MISTY1 and KASUMI are given in (2.3.3) and (2.3.4), respectively:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad (2.3.3)$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad (2.3.4)$$

CHAPTER 3

TEST CRITERIA FOR BOOLEAN FUNCTIONS

Shannon presented the principles of diffusion and confusion in 1949 [10]. To design a cipher according to the principle of diffusion means that one can design it to ensure that “the statistical structure of plaintext which leads to its redundancy is dissipated into long term statistics”. The higher the diffusion, the more output bits can be affected by a certain input bit.

To design a cipher according to the principle of confusion means that one can design it so as “to make the relation between the simple statistics of ciphertext and the simple description of key a very complex and involved one”. Ideally, every bit in the key influences every bit of the ciphertext and this dependence appears to be random. The security of cryptographic algorithms depends upon the strength, namely the diffusion and confusion properties of the constituting Boolean functions. In this chapter some test criteria for measuring cryptographic strength of Boolean functions are reviewed.

3.1 Boolean Functions [20]

A Boolean function produces a single bit result for each possible combination of values form many Boolean variables, namely a Boolean function of n variables is a function $f : F_2^n \to F_2$. A vector Boolean function $S(x) : F_2^n \to F_2^m$, where $n > 1$ and $m > 1$ maps n bits to m bits.

Definition 3.1 : If a Boolean function is in the form $f(x) = \sum_{i=1}^{n} a_i \cdot x_i \oplus c$, where $a_i, c \in F_2$ for $1 \le i \le n$, then $f(x)$ is called an affine function. $f(x)$ is called a linear function, $l_a(x)$, if $c = 0$. Using the dot product $a \bullet x$ of the vectors $a = (a_1,...,a_n)$ and $x = (x_1,...,x_n)$, $l_a(x) = a \bullet x$. The n-bit x also corresponds to an integer $0 \le x \le 2^n - 1$

The truth table of Boolean function $f(x):F_2^n \to F_2$ is found by evaluating $f(x)$ for all possible values of x, where x is ordered lexicographically, i.e., with respect to the ascending order of the integer represented by x.

Definition 3.2 : The Hamming distance between $f(x):F_2^n \to F_2$ and $g(x):F_2^n \to F_2$ is the function

$$d_H(f(x),g(x)) = \#\{x \in F_2^n \mid f(x) \neq g(x)\} \qquad (3.1)$$

and $0 \le d_H(f(x),g(x)) \le 2^n$.

Definition 3.3 : A Boolean function $f(x):F_2^n \to F_2$ is said to be balanced if its truth table contains as many 0's as 1's.

Definition 3.4 : The autocorrelation function of $f(x):F_2^n \to F_2$ is defined for all $d \in F_2^n$ as

$$r_f(d) = \sum_{x \in F_2^n} (-1)^{f(x)}(-1)^{f(x \oplus d)} \qquad (3.2)$$

Definition 3.5 : The Walsh-Hadamard transform of the Boolean function $f(x):F_2^n \to F_2$ is defined for all $a \in F_2^n$ as

$$W_f(a) = \sum_{x \in F_2^n} (-1)^{f(x)}(-1)^{a \bullet x} \quad , \qquad (3.3)$$

showing the correlation between $f(x)$ and the linear function, $l_a(x) = a \bullet x$.

Remark 3.1 : Let $g(x):F_2^n \to F_2$ and $f(x), g(x)$ have $d_H(f(x),g(x))$ different elements in their truth tables. By combining (3.1) and (3.3), we get

$$\sum_{x \in F_2^n} (-1)^{f(x)}(-1)^{g(x)} = \left(2^n - d_H(f(x),g(x))\right) + d_H(f(x),g(x))(-1) = 2^n - 2d_H(f(x),g(x))$$

Now, by replacing $g(x)$ with $l_a(x)$ one can obtain $W_f(a) = \sum_{x \in F_2^n} (-1)^{f(x)}(-1)^{l_a(x)} = 2^n - 2\mathrm{d_H}(f(x), l_a(x))$, therefore

$$d_H(f(x), l_a(x)) = 2^{n-1} - \frac{W_f(a)}{2} \qquad (3.4)$$

3.2 Cryptographic Properties of Boolean Functions

3.2.1 Completeness and Avalanche Criterion

The property of completeness was introduced by Kam and Davida [11]. If a cryptographic transformation is complete, then each output bit must depend on all of the input bits.

Definition 3.6 : Let $S(x): F_2^n \to F_2^n$. If for all $\{(i,j) \mid 1 \le i, j \le n\}$, there is at least one pair of input vectors $x, x_i \in F_2^n$ that differ in bit i, and $S(x)$ and $S(x_i)$ differ at least in bit j, then the vector Boolean function is complete.

Related to the autocorrelation function given by (3.2), the idea of avalanche effect was defined by Feistel [8]. For a given transformation to exhibit the avalanche effect, almost one half of the output bits should change whenever a single input bit is complemented. More formally, a vector Boolean function $S(x): F_2^n \to F_2^n$ satisfies the avalanche criterion if whenever an input bit is changed, half of the output bits change on the average.

Let $A^{e_i}(x) = S(x) \oplus S(x \oplus e_i) = (a_1^{e_i}(x), a_2^{e_i}(x), ..., a_n^{e_i}(x))$ be the avalanche vector for $e_i \in F_2^n$ such that $wt(e_i) = 1$. Then, the avalanche criterion is satisfied when the parameter defined in [1] as

$$AVAL(e_i) = \sum_{j=1}^{n} \sum_{x \in F_2^n} a_j^{e_i}(x) \tag{3.5}$$

is close to $n \cdot 2^{n-1}$ for all i, $1 \le i \le n$. $\overline{AVAL}(e_i) = \frac{1}{n \cdot 2^n} \sum_{j=1}^{n} \sum_{x \in F_2^n} a_j^i(x)$ is called the normalized avalanche parameter. If it is close to $1/2$ for all i, then $S(x): F_2^n \to F_2^n$ satisfies the avalanche criterion.

3.2.2 Strict Avalanche Criterion

The criteria of completeness and the avalanche effect were combined to define a new property called the strict avalanche criterion (SAC) by Webster and Tavares [19].

Definition 3.7 : Let $S(x): F_2^n \to F_2^n$. Consider the input vectors $x, x_i \in F_2^n$ that differ only in bit i, $1 \le i \le n$. Then, $A^{e_i}(x) = S(x) \oplus S(x_i)$. If S is to meet SAC, the probability that each bit in $A^{e_i}(x)$ is equal to 1 should be one half over the set of all possible input vectors, $x, x_i \in F_2^n$, for all values of i.

If a vector Boolean function, $S(x): F_2^n \to F_2^n$, is to satisfy the Strict Avalanche Criterion, the change of the i^{th} input bit results in the change of the j^{th} output bit exactly for half of the input vectors, so the probability that the j^{th} output bit is complemented is ${}^1\!/_2$.

By using the autocorrelation function, defined by (3.2) one can express the Strict Avalanche Criterion (SAC) as follows :

Definition 3.8 : Let $x \in F_2^n$ and $f(x): F_2^n \to F_2$ be a Boolean function with the auto correlation function $r_f(e_i)$. $f(x)$ satisfies the SAC if $r_f(e_i) = \sum_{x \in F_2^n} (-1)^{f(x)} (-1)^{f(x \oplus e_i)} = 0$ for all $e_i \in F_2^n$ such that $wt(e_i) = 1$, where $1 \le i \le n$. Namely, for the Boolean function which satisfy SAC, $f(x) \oplus f(x \oplus e_i)$ is balanced for all $e_i \in F_2^n$ such that $wt(e_i) = 1$, where $1 \le i \le n$.

The original definition of SAC can be extended to an arbitrary input difference vector $i \in F_2^n - \{0\}$ Let $S(x): F_2^n \to F_2^n$ be a vector Boolean function and $A^i(x) = S(x) \oplus S(x \oplus i) = (a_1^i(x), a_2^i(x), \ldots, a_n^i(x))$ be the avalanche vector for any $i \in F_2^n - \{0\}$. Then, if

$$SAC(i, j) = \sum_{x \in F_2^n} a_j^i(x) \tag{3.6}$$

is close to 2^{n-1} for all $1 \le j \le n$, then the vector Boolean function, $S(x): F_2^n \to F_2^n$, satisfies the SAC.

The measure D_j^i, normalized distance to SAC, is defined in [1] and it can be used to indicate how close the vector Boolean function $S(x)$ is to satisfy SAC.

$$D_j^i = \frac{1}{2^{n-1}}\left(2^{n-1} - \sum_{x \in F_2^n} a_j^i\right) \tag{3.7}$$

If the strict avalanche criteria is exactly satisfied, then $\left|D_j^i\right| = 0$ for all output bits. In the worst case $\left|D_j^i\right| = 1$. If SAC is satisfied, then the completeness and avalanche criterion are also satisfied. However, the satisfaction of the avalanche criterion does not ensure that SAC is satisfied.

3.2.3 Nonlinearity

Nonlinearity is one of the most critical indicators of the cryptographic strength of a Boolean function.

Definition 3.9 : The nonlinearity of a Boolean function $f(x): F_2^n \to F_2$ is the minimum distance of $f(x)$ to the set of affine functions [20].

$$N_f = \min_{a,c} d_H(f(x), (a \bullet x \oplus c)) = \min_a \{d_H(f(x), l_a(x)), d_H(f(x), \bar{l}_a(x))\}, \tag{3.8}$$

where $\bullet$ denotes the dot product, $l_a(x)$ is a linear function and $\bar{l}_a(x) = l_a(x) \oplus 1$.

Using (3.4) in (3.8)

$$N_f = \min_a \{d_H(f(x), l_a), d_H(f(x), \bar{l}_a)\}$$

$$= \min_a \{2^{n-1} - \frac{1}{2}\sum_{x \in F_2^n} (-1)^{f(x)} \cdot (-1)^{l_a(x)}, 2^{n-1} - \frac{1}{2}\sum_{x \in F_2^n} (-1)^{f(x)} \cdot (-1)^{\bar{l}_a(x)}\}$$

$$= 2^{n-1} - \frac{1}{2}\max_{a \in F_2^n} | W_f(a) |, \qquad (3.9)$$

where $W_f(a)$ is the Walsh-Hadamard transform of the Boolean function $f(x)$ given by Definition 3.5.

The nonlinearity of a Boolean function $f(x): F_2^n \to F_2$ can be extended to vector Boolean functions $S(x): F_2^n \to F_2^n$, $S(x) = (f_1(x),..., f_n(x))$ by $N_S = \min_{c \in F_2^n - \{0\}} \{N_f \mid c \bullet S(x)\}$, where $c \in F_2^n - \{0\}$ is called a masking vector.

3.2.4 Linear Approximation Table (LAT) Distribution

Linear cryptanalysis [12] is a known plaintext attack that is based on effective linear approximate relations between the plaintext, ciphertext and the key. The linear approximation tables of the vector Boolean functions which constitute the block cipher are exploited for this purpose. LAT is an important tool to measure the security of s-boxes against linear cryptanalysis.

Definition 3.7 : Let $x, y \in F_2^n$ and $S(x): F_2^n \to F_2^n$. Each element of the Linear Approximation Table is defined as

$$\underset{a,c \in F_2^n}{LAT}(a,c) = \#\ \{x \mid c \bullet S(x) = a \bullet x\} - 2^{n-1} = \#\ \{x \mid c \bullet S(x) = a \bullet x\} - 2^{n-1} \qquad (3.10)$$

$$= \#\{x \mid c \bullet S(x) = l_a(x)\} - 2^{n-1}$$

$$= 2^n - d_H(c \bullet S(x),\ l_a(x)) - 2^{n-1}$$

$$= 2^{n-1} - d_H(c \bullet S(x),\ l_a(x)) \qquad (3.11)$$

where a and c are respectively the row and column indices and $\bullet$ denotes the dot product of vectors.

Using (3.4), the distance between $c \bullet S(x)$ and $l_a(x)$, and (3.11), we have

$$d_H(c \bullet S(x),\ l_a(x)) = 2^{n-1} - \frac{1}{2} W_{c \bullet S(x)}(a) = 2^{n-1} - \underset{a,c \in F_2^n}{LAT}(a,c), \text{ and}$$

$$\underset{a,c \in F_2^n - \{0\}}{LAT}(a,c) = \frac{W_{c \bullet S(x)}(a)}{2}.$$

Remark 3.2 : By combining nonlinearity (see 3.9) and LAT, we obtain

$$\max_{a,c\in F_2^n-\{0\}} |LAT(a,c)| = 2^{n-1} - N_f \qquad (3.12)$$

An arbitrary linear combination of the output bits, $c \bullet S(x)$ is most correlated with an arbitrary linear combination of the input bits, whenever $d_H(c \bullet S(x),\ a \bullet x) = 2^{n-1}$. If this distance is 0, maximum positive correlation occurs. Similarly, if this distance is 2^n, $c \bullet S(x)$ and $a \bullet x$ have maximum negative correlation. Therefore, the $\underset{a,c\in F_2^n}{LAT}(a,c)$ defined by (3.11) also measures how close the two functions $c \bullet S(x)$ and $a \bullet x$ are to the ideal situation of being uncorrelated. Normalized values $\underset{a,c\in F_2^n}{LAT}(a,c)/2^n$ can also be considered as the bias of the probability $P\{c \bullet S(x) = a \bullet x\}$ from the ideal probability of ½. The reason as follows:

$$P\{c \bullet S(x) \neq a \bullet x\} = \frac{d_H(c \bullet S(x), a \bullet x)}{2^n}$$

$$2^n P\{c \bullet S(x) \neq a \bullet x\} = 2^n - \#\{x \mid c \bullet S(x) = a \bullet x\}$$

$$2^n \left(1 - P\{c \bullet S(c) = a \bullet x\}\right) = 2^{n-1} - \frac{W_{c\bullet S(x)}(a)}{2}$$

$$P\{c \bullet S(x) = a \bullet x\} = \frac{1}{2} + \frac{W_{c\bullet S(x)}}{2^{n+1}} = \frac{1}{2} + \frac{\underset{a,c\in F_2^n}{LAT}(a,c)}{2^n}$$

Hence, the bias of the probability $P\{c \bullet S(x) = a \bullet x\}$ is

$$P\{c \bullet S(x) = a \bullet x\} - \frac{1}{2} = \frac{\underset{a,c\in F_2^n}{LAT}(a,c)}{2^n} \qquad (3.13)$$

Probability bias (3.12) varies in the interval [-1/2, 1/2] since $\max_{a,c\in F_2^n-\{0\}} |LAT(a,c)| = 2^{n-1}$. Large elements of LAT are not desired since they indicate high probability of linear relations between the input and the output.

3.2.5 Exclusive or –XOR Table Distribution

Differential cryptanalysis [2] is a chosen plaintext attack, which uses the propagation of input differences to output differences in iterated transforms. In other words, it exploits the high propagation probability of certain occurrences of plaintext differences to the last round input difference of the cipher. Main part of the differential cryptanalysis is making XOR tables of the vector Boolean functions. XOR table contains the distribution on the differential output.

Definition 3.8 : Let $x, y \in F_2^n$ and $S(x): F_2^n \to F_2^n$. Let two inputs to the system be x', x'' with the corresponding outputs y', y'' respectively. The input and output differences are given by $\Delta x = x' \oplus x''$ and $\Delta y = y' \oplus y''$, respectively. Then, XOR table can be constructed by using

$$XOR(\Delta x, \Delta y) = \#\{x \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\} \tag{3.14}$$

The rows of the matrix, Δx, represent the change in the output of the s-box. The sum of all values in a row or a column is 2^n. The parameter $\max\limits_{\Delta x, \Delta y \neq 0} XOR(\Delta x, \Delta y) = \delta$ is called differential uniformity. XOR table of an s-box gives information about the security of the block cipher against differential cryptanalysis. If differential uniformity is large, this is an indication of an insecure block cipher.

The main difference from LAT is that XOR table distribution involves comparing XOR of the two inputs to the XOR of the corresponding outputs. In LAT, we try to find linear relationship between a subset of input and output bits.

One can find resemblances between the definition of SAC and XOR table distribution. SAC is useful to get an idea when input bits are changed how often output bits are affected. XOR table represents when the input bits are changed, the number of occurrences of the corresponding output difference for given input difference. In other words, in XOR table one finds the number of output differences, in SAC one calculates the number of 1’s in each bit column for all output differences.

Mentioned criteria are used to test the s-boxes of block ciphers. They are not very practical for application to the overall cipher. Due to this reason, Chapter 5 is organized to test in terms of avalanche and randomness criterion the overall cipher.

CHAPTER 4

S-BOX TEST RESULTS FOR MISTY1, KASUMI AND RIJNDAEL

The resistance of block ciphers to cryptanalytic attacks depends heavily on their diffusion and confusion properties. The overall nonlinearity of a cipher is usually provided by the s-boxes, which should be chosen carefully.

In this chapter, we investigate the cryptographic strength of the s-boxes of MISTY1, KASUMI, and RIJNDAEL, in terms of linear approximation table, XOR table distributions and the strict avalanche criterion. These s-boxes are of sizes 7x7 and 9x9 that are described in section 2.4, 8x8 s-box of RIJNDAEL explained in section 4.4.1. Then, we consider the 16x16 FI function used in MISTY1 and KASUMI to find its strict avalanche characteristics. We finally replace the s-boxes in the FI function of KASUMI with RIJNDAEL's s-box. Calling the new cipher KASUMI-R, we compare the FI functions of MISTY1, KASUMI and KASUMI-R in terms of their strict avalanche characteristics.

4.1 Test Results of LAT Distribution

The LAT is a matrix of size 128x128 for 7x7 s-boxes, 256x256 for 8x8 s-boxes and 512x512 for 9x9 s-boxes, whose elements are calculated by

$$\underset{a,c \in F_2^n}{LAT(a,c)} = \#\ \{x \mid c \bullet S(x) = a \bullet x\} - 2^{n-1} \text{ (see 3.11).}$$

Since the size of LAT for these s-boxes is very large, we only present some part of the LAT's corresponding to input and output differences weight one. We then compute the nonlinearity measure of each s-box by (see 3.12), i.e., $N_f = 2^{n-1} - \max_{a,c \in F_2^n - \{0\}} \mid LAT(a,c) \mid$. However, the maximum entries encountered in these partial LAT's are observed to be the same as the maxima of the overall LAT's for each s-box.

4.1.1 Results for the 7x7 S-boxes of MISTY1 and KASUMI

When we investigate LAT's of MISTY1's and KASUMI's 7x7 s-boxes, it is seen that there are only three values {-8, 0, 8}. The number of 8's is 4060 and the number of -8's is 4068 for both of the s-boxes. So, one can say that there is no significant difference between these s-boxes. As an interesting observation we note that the number of LAT elements, different from zero is $2^{2n-2} - \left(2^{n-2} \mp \frac{2^{\frac{n-1}{2}}}{2} \right)$ for both s-boxes and $\max_{a,c \in F_2^n} |LAT(a,c)| = 2^{\frac{n-1}{2}}$. This gives the nonlinearity of 56 for 7x7 s-boxes, by using (3.12). Table 4.1 and Table 4.2 show LAT distributions for the s-boxes of MISTY1 and KASUMI for single bit input and output differences.

Table 4.1. LAT Distribution of MISTY1's 7x7 S-box
for Single bit Input and Output Differences

Output Sum	1	2	4	8	16	32	64
Input Sum							
1	-8	0	8	0	0	0	0
2	0	0	0	-8	-8	0	8
4	0	-8	8	0	0	8	8
8	0	-8	0	8	0	8	8
16	-8	0	8	0	0	-8	0
32	0	0	0	-8	-8	0	0
64	0	-8	0	0	0	8	8

Table 4.2. LAT Distribution of KASUMI's 7x7 S-box
for Single bit Input and Output Differences

Output Sum	1	2	4	8	16	32	64
Input Sum							
1	8	0	-8	0	-8	0	0
2	0	-8	0	8	0	0	0
4	-8	0	0	0	0	-8	8
8	8	8	-8	0	-8	0	0
16	8	0	-8	0	-8	-8	0
32	0	-8	0	8	-8	0	0
64	0	0	0	0	0	-8	8

4.1.2 Results for the 9x9 S-boxes of MISTY1 and KASUMI

There are only three values {-16, 0, 16} in the LAT's of the 9x9 s-boxes. This gives the nonlinearity of 240 for 9x9 s-boxes. Table 4.3 and Table 4.4 show partial LAT's for 9x9 s-boxes of MISTY1 and KASUMI. Comparing the two tables, one observes a single |16| in each row and column for KASUMI's 9x9 s-box, which is not the case for the first column of Table 4.3. This means that the first output bit of 9x9 s-box of MISTY1 is correlated with all 9-bit unit vectors as opposed to KASUMI's s-box. However, this correlation is not large ($-16/256 = -6.25\%$). Regarding the whole 512x512 LAT's, the number of 16's is 65400 and the number of -16's 65416 for both of the s-boxes. We again observe that the number of LAT elements, different from zero is $2^{2n-2} - \left(2^{n-2} \mp \frac{2^{\frac{n-1}{2}}}{2}\right)$ and $\max\limits_{a,c\in F_2^n} |LAT(a,c)| = 2^{\frac{n-1}{2}}$.

Table 4.3. LAT Distribution of MISTY1's 9x9 S-box
for Single bit Input and Output Differences

Output Sum	1	2	4	8	16	32	64	128	256
Input Sum									
1	-16	0	0	16	0	0	0	0	0
2	-16	0	0	0	16	0	0	0	0
4	-16	0	0	0	0	16	0	0	0
8	-16	0	0	0	0	0	-16	0	0
16	-16	0	0	0	0	0	0	0	-16
32	-16	0	0	0	0	0	0	-16	0
64	-16	0	0	0	0	0	0	0	0
128	-16	-16	0	0	0	0	0	0	0
256	-16	0	16	0	0	0	0	0	0

Table 4.4. LAT Distribution of KASUMI's 9x9 S-box
for Single bit Input and Output Differences

Output Sum	1	2	4	8	16	32	64	128	256
Input Sum									
1	0	0	0	0	0	0	16	0	0
2	0	0	16	0	0	0	0	0	0
4	0	0	0	0	0	-16	0	0	0
8	-16	0	0	0	0	0	0	0	0
16	0	0	0	0	16	0	0	0	0
32	0	0	0	16	0	0	0	0	0
64	0	-16	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0	16
256	0	0	0	0	0	0	0	-16	0

4.1.3 Results for the 8x8 S-box of RIJNDAEL

In RIJNDAEL's 8x8 s-box, as opposed to previous cases, there are various LAT values differing between -16 and 16 when we consider the whole LAT. This gives the nonlinearity of 112 by using (3.12). The number of occurrences of $(2^{16}-1)$ LAT elements is shown in Figure 4.1.

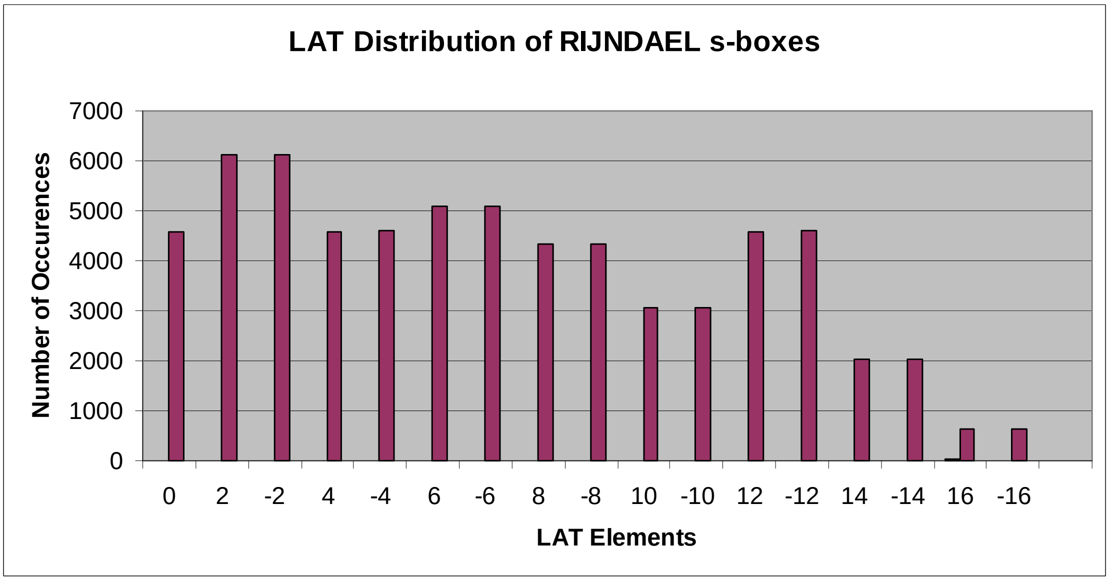

Figure 4.1 Number of Occurrences of LAT Elements for RIJNDAEL S-box

In Table 4.5 LAT distribution of RIJNDAEL's 8x8 s-box is demonstrated for input and output differences of single weight.

Table 4.5. LAT Distribution of RIJNDAEL's 8x8 S-box for Single bit Input and Output Differences

Output Sum	1	2	4	8	16	32	64	128
Input Sum								
1	12	0	14	12	8	-4	4	12
2	2	8	2	6	-2	8	-16	-2
4	-8	2	6	6	12	-16	2	-2
8	2	2	4	0	12	6	2	4
16	-12	-2	-6	-2	-8	-10	0	8
32	6	-10	-2	-12	2	0	-8	12
64	4	-4	-12	16	6	-8	-12	4
128	-12	-12	16	14	-8	-12	-4	-4

4.1.4 Comparison of MISTY, KASUMI and RIJNDAEL S-boxes

The main difference between LAT's of 7x7, 8x8 and 9x9 s-boxes is that, RIJNDAEL's 8x8 s-box has 17 different LAT values, whereas MISTY1's and

KASUMI’s s-boxes have only 3 different values. Table 4.6 shows the distribution percentages of LAT elements for the three s-boxes.

Table 4.6. Distribution Percentages of LAT Elements for the Three S-boxes

S-box Size	**7×7**	**8×8**	**9×9**
Percentages of			
0’s	0.50	0.07	0.50
|2|’s	0	0.18	0
|4|’s	0	0.14	0
|6|’s	0	0.15	0
|8|’s	0.49	0.13	0
|10|’s	0	0.09	0
|12|’s	0	0.14	0
|14|’s	0	0.06	0
|16|’s	0	0.01	0.49

Maximum bias of the probability $P\{a \bullet x = c \bullet S(x)\}$ from ½, $\frac{\max_{a,c \in F_2^n - \{0\}} |LAT(a,c)|}{2^n}$, is calculated for the three s-boxes, and given in Table 4.7. The maximum probability biases for the 7x7 and 8x8 s-boxes are found to be same. On the other hand, 9x9 s-box has the best probability bias as observed in Table 4.7.

Table 4.7. Maximum Probability Biases for the Three S-boxes

S-box Size	**7×7**	**8×8**	**9×9**
Probability Bias	1/16	1/16	1/32

Nonlinearity, given by $N_f = 2^{n-1} - \max_{a,c \in F_2^n - \{0\}} |LAT(a,c)|$ (see 3.12), of the three s-boxes is shown in Table 4.8.

Table 4.8. Nonlinearity for the Three S-boxes

S-box Size	7×7	8×8	9×9
Nonlinearity	56	112	240

4.2 Test Results of XOR Table Distribution

The XOR table is a matrix of size 128x128 for the 7x7 s-boxes, 256x256 for the 8x8 s-boxes and 512x512 for the 9x9 s-boxes, whose elements are calculated by

$$XOR(\Delta x, \Delta y) = \#\{x \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\} \text{ (see 3.14),}$$

where $S(x): F_2^n \to F_2^n$, $x, \Delta x, \Delta y \in F_2^n$ and $\Delta x, \Delta y$ are the input and output differences, respectively. Because of the same reason of LAT, the size of XOR table for s-boxes is very large, we only give partial XOR tables corresponding to input and output differences of single weight.

4.2.1 Results for the 7x7 S-boxes of MISTY1 and KASUMI

When we compare the 7x7 s-boxes of MISTY1 and KASUMI according to their XOR table distributions, XOR tables contain only two values {0, 2}. Considering the overall XOR tables, the number of 0's and 2's except the first row is equal and can be defined as 2^{n-1}. Then, one can say that there is no significant difference between these s-boxes. Table 4.9 and Table 4.10 show XOR table distributions for the s-boxes of MISTY1 and KASUMI for single bit input and output differences.

Table 4.9. XOR Table Distribution of MISTY1's 7x7 S-box for Single bit Input and Output Differences

Δy Δx	1	2	4	8	16	32	64
1	2	0	0	0	0	0	0
2	0	0	0	2	0	0	0
4	0	0	0	0	0	2	0
8	0	0	0	0	0	2	2
16	0	0	2	0	0	2	2
32	0	0	2	0	2	2	2
64	0	2	2	0	2	2	2

Table 4.10. XOR Table Distribution of KASUMI's 7x7 S-box for Single bit Input and Output Differences

Δy Δx	1	2	4	8	16	32	64
1	2	0	2	2	2	2	0
2	2	0	0	2	2	2	0
4	2	0	0	0	2	2	0
8	2	0	0	0	2	0	0
16	2	0	0	0	0	0	0
32	0	2	0	0	0	0	0
64	0	0	0	0	0	0	2

4.2.2 Results for the 9x9 S-boxes of MISTY1 and KASUMI

Similar to previous case, there are only two values {0, 2} in their XOR tables. Moreover, we again observe that regarding the whole 512x512 XOR tables the number of 0's and 2's is equal for both s-boxes as 2^{n-1}. XOR table distributions of MISTY1 and KASUMI's 9x9 s-boxes for input and output differences of single weight are shown in Table 4.11 and Table 4.12. It is observed that there is only one 2 in each row and column for KASUMI's 9x9 s-box which is not the case for the seventh row of Table 4.11. The seventh input difference of 9x9 s-box of MISTY1 is correlated with all output differences except the first one in Table 4.11 on the contrary to the s-box of KASUMI. However, this correlation is not large.

Table 4.11. XOR Table Distribution of MISTY1's 9x9 S-box for Single bit Input and Output Differences

Δy / Δx	1	2	4	8	16	32	64	128	256
1	0	0	0	2	0	0	0	0	0
2	0	0	0	0	2	0	0	0	0
4	0	0	0	0	0	2	0	0	0
8	0	0	0	0	0	0	2	0	0
16	0	0	0	0	0	0	0	0	2
32	0	0	0	0	0	0	0	2	0
64	0	2	2	2	2	2	2	2	2
128	0	2	0	0	0	0	0	0	0
256	0	0	2	0	0	0	0	0	0

Table 4.12. XOR Table Distribution of KASUMI' 9x9 S-box for Single bit Input and Output Differences

Δy / Δx	1	2	4	8	16	32	64	128	256
1	0	0	0	0	0	0	2	0	0
2	0	0	2	0	0	0	0	0	0
4	0	0	0	0	0	2	0	0	0
8	2	0	0	0	0	0	0	0	0
16	0	0	0	0	2	0	0	0	0
32	0	0	0	2	0	0	0	0	0
64	0	2	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0	2
256	0	0	0	0	0	0	0	2	0

4.2.3 Results for the 8x8 S-box of RIJNDAEL

In RIJNDAEL’s 8x8 s-box, as opposed to the previous cases, there are three XOR values differing between 0 and 4. Considering overall XOR table, every row and column except the first row and column contains exactly one 4. XOR table distribution of RIJNDAEL’s 8x8 s-box for single bit input and output differences is demonstrated in Table (4.13). It is noticed that the value of 4 does not appear in these specific rows and columns.

Table 4.13. XOR Table Distribution of RIJNDAEL’s 8x8 S-box for Single bit Input and Output Differences

Δy Δx	1	2	4	8	16	32	64	128
1	0	0	0	0	0	0	0	0
2	2	0	2	2	0	0	0	2
4	0	2	2	2	0	0	2	0
8	2	0	0	2	2	0	0	2
16	0	2	2	2	0	0	0	2
32	0	2	0	2	0	0	2	0
64	0	0	2	0	2	0	2	2
128	0	0	0	2	2	2	2	2

4.2.4 Comparison of MISTY1, KASUMI and RIJNDAEL S-boxes

There are two significant differences between XOR tables of 7x7, 8x8 and 9x9 s-boxes. The first one is differential uniformities of s-boxes and the second difference is related with the number of elements for each row of the s-box. Differential uniformity defined by, $\delta = \max\limits_{\Delta x, \Delta y \neq 0} XOR(\Delta x, \Delta y)$, is 4 for RIJNDAEL’s, whereas it is 2 for MISTY1’s and KASUMI’s s-boxes.of the 7x7, 8x8 and 9x9 s-boxes, as shown in Table 4.14.

Table 4.14. Differential Uniformity of XOR Table Elements for Each Row of the Three S-boxes

S-box Size	7×7	8×8	9×9
Differential Uniformity	2	4	2

The number of XOR table 0's and 2's is equal for MISTY1's and KASUMI's as opposed to RIJNDAEL's s-box. Table 4.15 summarizes the number of XOR table elements in each row (except the first row).

Table 4.15. XOR Table Elements for Each Row of the Three S-boxes

S-box Size	7×7	8×8	9×9
Number of			
0's	2^{n-1}	$2^{n-1}+1$	2^{n-1}
2's	2^{n-1}	$2^{n-1}-2$	2^{n-1}
4's	0	1	0

4.3 Test Results of Strict Avalanche Criterion for S-boxes

SAC values constitute a table of size 127x7 for the 7x7 s-boxes, 255x8 for the 8x8 s-box and 511x9 for the 9x9 s-boxes, whose elements are calculated by

$$D_j^i = \frac{1}{2^{n-1}}\left(2^{n-1} - \sum_{x\in F_2^n} a_j^i\right) \quad \text{(see 3.7),}$$

where i is any n bit vector, $1 \le i \le 2^n - 1$, a_j^i is the j^{th} avalanche variable, $1 \le j \le n$. In the following, we sketch D_j^i versus j for each s-box, where the curves corresponding to different values of $i \in F_2^n - \{0\}$ are drawn on the top of each other.

4.3.1 Results for the 7x7 S-boxes of MISTY1 and KASUMI

For the 7x7 s-boxes, D_j^i curves given by (3.7) where $i \in F_2^7 - \{0\}$ are depicted in Figure 4.2 and Figure 4.3 for MISTY1 and KASUMI, respectively. In the figure there are 127 curves corresponding to all input differences 1-127 and each curve is shown in a different color. The maximum of the normalized distance to SAC over all i and j is found as $|D|_{\max} = 0.125$ with corresponding values of $i = 127$. It is observed that for these s-boxes SAC gets this highest value for each bit a_j^{127} of the avalanche vector, that is for $j = 1,2,...,7$. However, corresponding deviation from ideal randomness is small i.e., 12.5%. When KASUMI and MISTY1 7x7 s-boxes are compared, there is no distinctive property between them. All values in Figure 4.2 and Figure 4.3 are equal to 0 or 0.125.

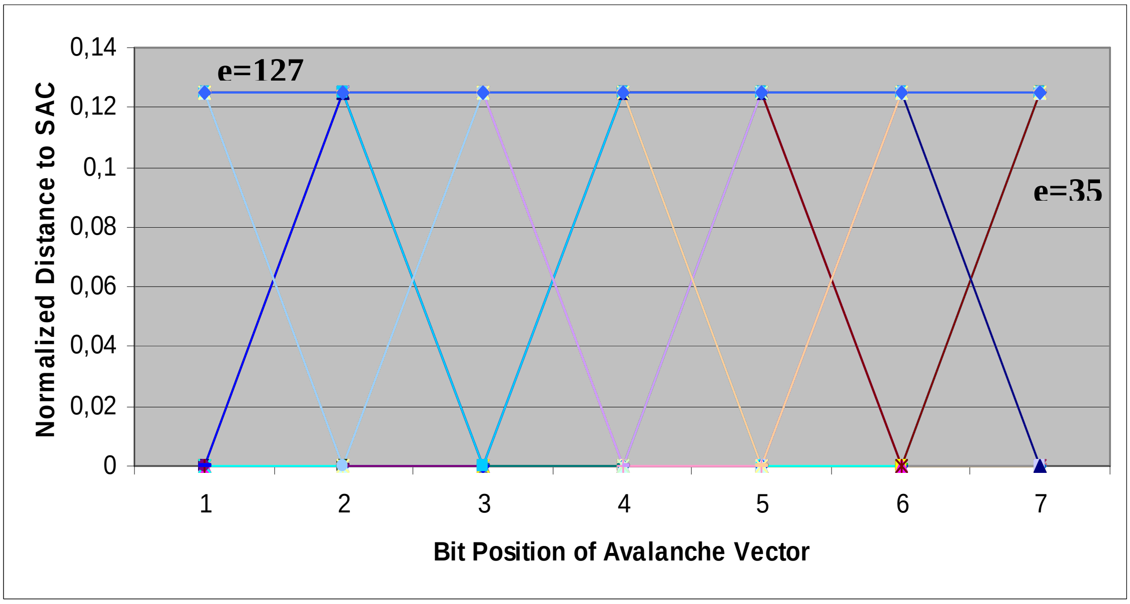

Figure 4.2 Normalized Distance to SAC for the 7 x 7 S-box of MISTY1

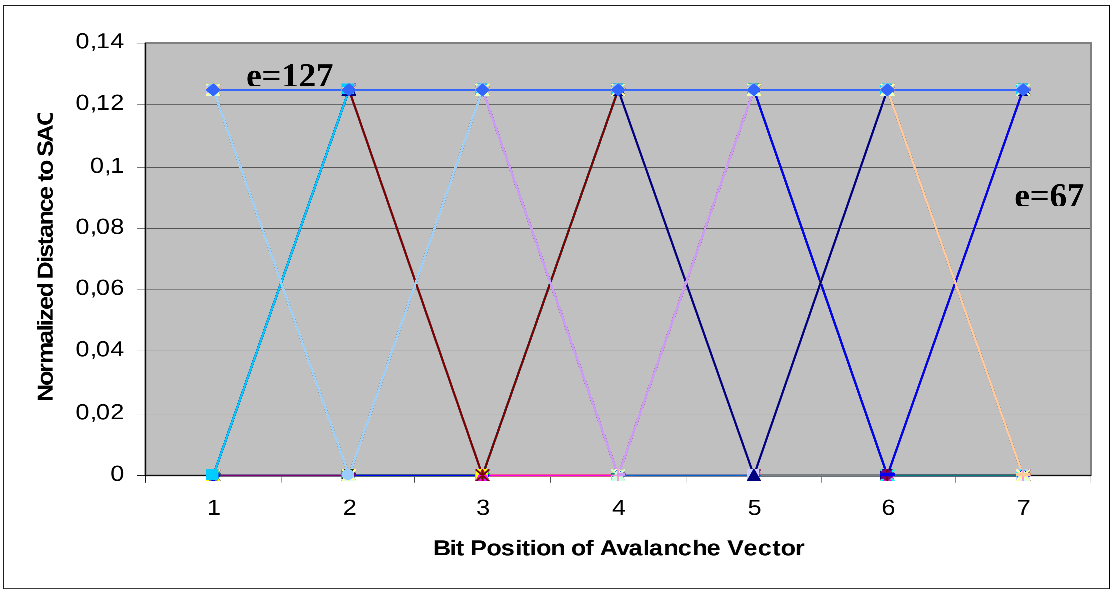

Figure 4.3 Normalized Distance to SAC for the 7 x 7 S-box of KASUMI

4.3.2 Results for the 9x9 S-boxes of MISTY1 and KASUMI

Figure 4.4 and Figure 4.5 show D_j^i curves given by (3.7) where $i \in F_2^9 - \{0\}$ for the 9x9 s-boxes. In the figures different colors correspond to various input differences 1-511. The maximum of normalized distance to SAC is found as $|D|_{\max} = 1$ with corresponding values of $i = 128$ and $j = 8$ for MISTY1, and $i = 128$ and $j = 1$, where j is the bit position of avalanche vector. By this way we can say that both of them do not satisfy the strict avalanche criterion. When 9x9 s-boxes of KASUMI and MISTY1 are compared in terms of SAC, there is only one difference between them. MISTY1's 9x9 s-box for input difference of 173 and 429, and seventh bit of avalanche vector has $|D_7^{173}| = |D_7^{429}| = 0.5$. In Figure 4.4 we have three different values {0, 0.5, 1}. However, all of the values in Figure 4.5 are equal to 0 or 1. An interesting observation is that in KASUMI 9x9 s-box all input differences whose weights are different from 1, i.e., $wt(i) \neq 1$ give $|D_j^i| = 0$.

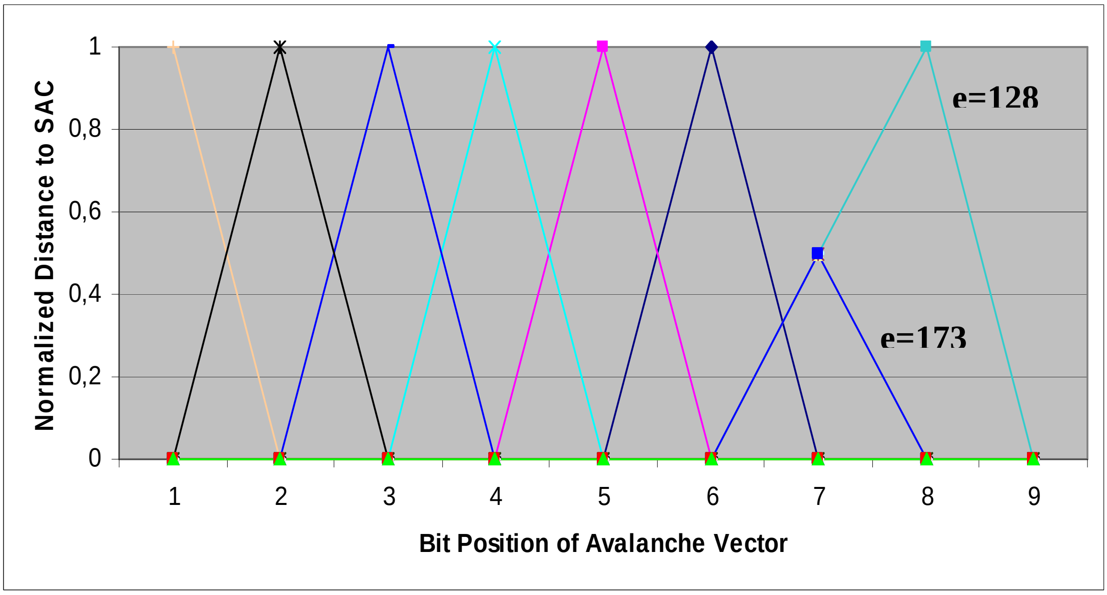

Figure 4.4 Normalized Distance to SAC for the 9 x 9 S-box of MISTY1

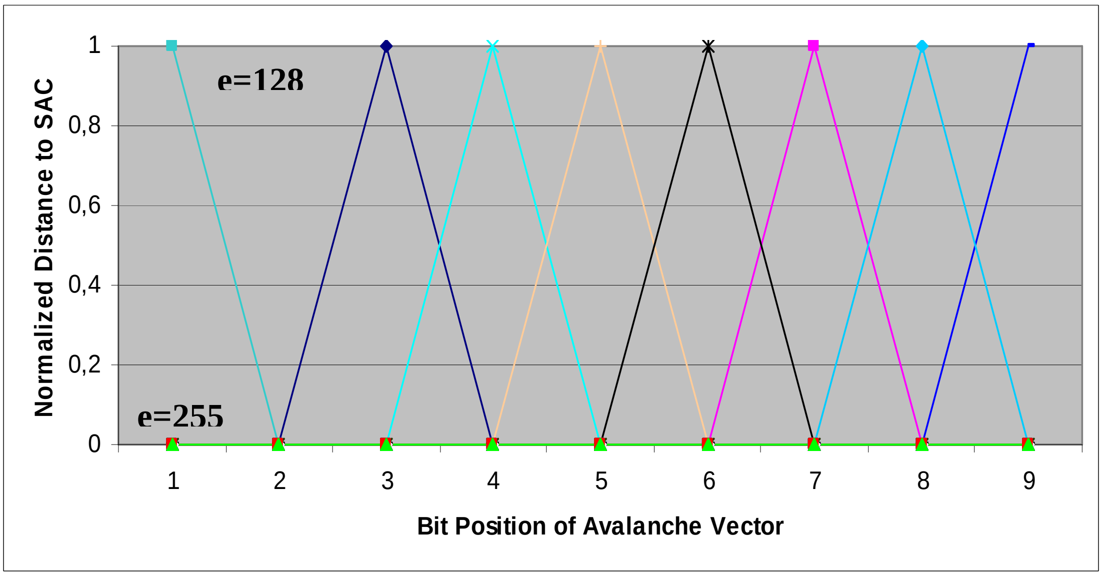

Figure 4.5 Normalized Distance to SAC for the 9 x 9 S-box of KASUMI

4.3.3 Results for the 8x8 S-box of RIJNDAEL

For the 8x8 s-box of RIJNDAEL, D_j^i curves given by (3.7) are sketched versus j in Figure 4.6 where different colors correspond to various values of $i \in F_2^8 - \{0\}$. The highest normalized distance to SAC is obtained $\left|D_j^i\right| = 0.125$ for input difference of 72 and $j = 6{,}7$. SAC gets the highest value for at most two bits of avalanche vector for Rijndael 8x8 s-box. Although RIJNDAEL does not satisfy strict avalanche criteria, its normalized distance to SAC values are very near to 0, like 7x7 s-boxes of MISTY1 and KASUMI, which is quite good.

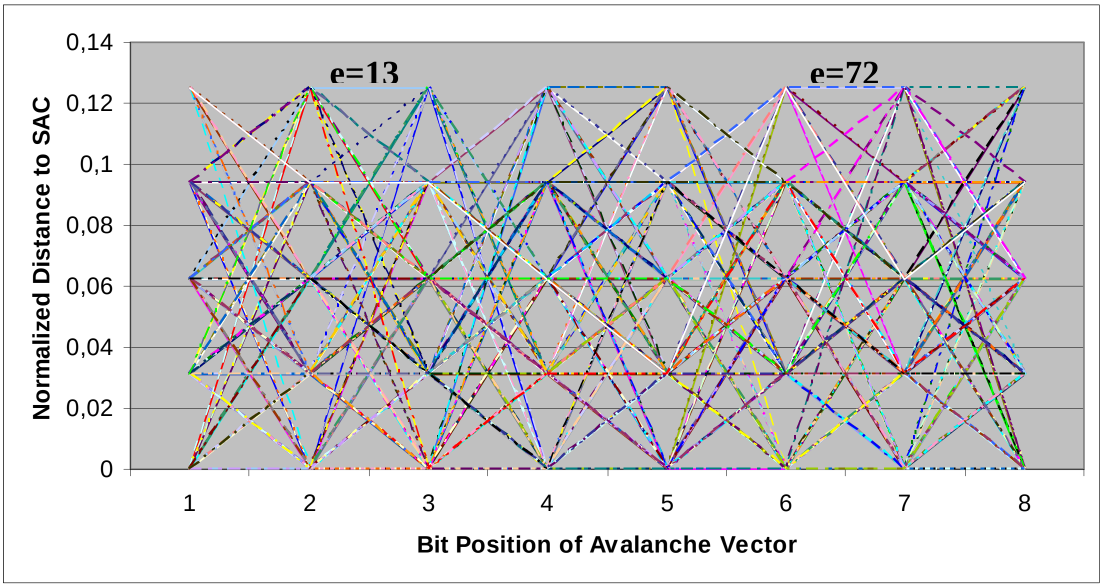

Figure 4.6 Normalized Distance to SAC of RIJNDAEL 8 x 8 S-box

4.3.4 Comparison of MISTY1, KASUMI and RIJNDAEL S-boxes

When the s-boxes of KASUMI, MISTY1 and RIJNDAEL are compared in terms of SAC, although most of the normalized distance to SAC values is 0 in 9x9 s-boxes of MISTY1 and KASUMI yield the most undesirable value of $\left|D_j^i\right|_{\max} = 1$, on the other hand, in RIJNDAEL all values are around 0. Table 4.16 summarizes the obtained results. In Table 4.16 only one value is given in the corresponding i column.

Table 4.16 Normalized Distance to SAC for the S-boxes

S-box	$\left\|D_j^i\right\|_{\max}$	Corresponding i
MISTY 7x7 s-box	0.125	127
KASUMI 7x7 s-box	0.125	127
MISTY 9x9 s-box	1	128
KASUMI 9x9 s-box	1	128
RIJNDAEL 8x8 s-box	0.125	72

4.4 Test Results of Strict Avalanche Criterion for FI Functions

In this section, we compare the FI functions of MISTY1 and KASUMI in terms of SAC. Then, we define a new FI function by replacing both s-boxes of KASUMI by RIJNDAEL’s s-box in subsection 4.4.1. Then, we give the SAC curves for each FI function.

4.4.1 FI Function with RIJNDAEL S-box

By replacing s-boxes of KASUMI with RIJNDAEL’s s-box a new FI function is obtained. The s-box of RIJNDAEL was designed the inversion function $x \to x^{-1}$ in F_2^8. This function is constructed according to [14] since it provides good differential and linear properties as a nonlinear transformation. Firstly, the inverse of the element in F_2^8 is found for every nonzero element. Then, the resulting inverse is transformed by an affine transformation to produce the output. Construction matrix is as follows:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad (4.4.1)$$

where x_i is the i^{th} bit of the multiplicative inverse in the finite field F_2^8 for $0 \le i < 8$. Calling an element $x \in F_2^8$, $x = (x_0 ... x_8)$ the s-box output is $y = A \cdot x + c$ with $c = (c_0 ... c_8)$ if x is equal to the inverse of the s-box input.

The FI function with RIJNDAEL’s s-box, shown in Figure 4.7, also takes a 16-bit input P and a 16-bit subkey KI_{ij} and it gives 16-bit output C.

The input P is split this into two equal components, where $P = L_0 \parallel R_0$.

Similarly, the subkey KI_{ij} is split into two equal components, where $KI_{ij} = KI_{ij1} \parallel KI_{ij2}$. Then, the modified FI function is defined as follows :

$$
\begin{aligned}
R_1 &= S8[L_0] \oplus R_0 \\
L_1 &= R_0 \\
R_2 &= S8[L_1] \oplus R_1 \oplus KI_{ij1} \\
L_2 &= R_1 \oplus KI_{ij2} \\
R_3 &= S8[L_2] \oplus R_2 \\
L_3 &= R_2 \\
R_4 &= R_3 \\
L_4 &= S8[L_3] \oplus R_3
\end{aligned}
\qquad (4.4.1)
$$

$$C = L_4 \parallel R_4 \qquad (4.4.2)$$

Main difference between KASUMI's FI function is the lack of zero-extend (ZE) and truncate (TR) functions, since the input is split into two equal components.

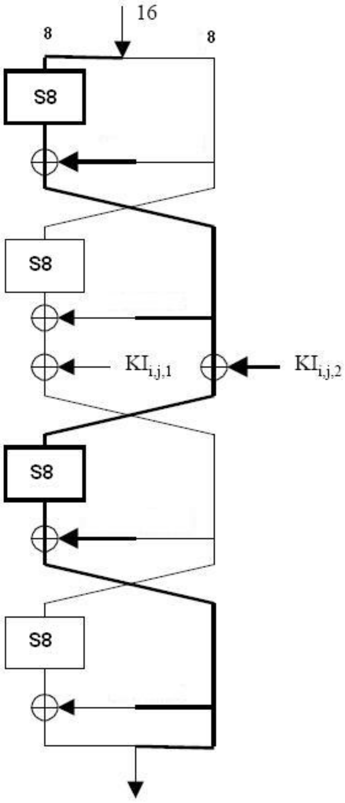

Figure 4.7 FI function of KASUMI-R

In the following sections we give the normalized distance to SAC, $\left|D_j^i\right|$, curves for the three FI functions. We consider 32 curves to all possible one-bit and some two-bit input differences which are $i \in \{e_1, ..., e_{16}, e_1 \oplus e_2, e_2 \oplus e_3, ..., e_{15} \oplus e_{16}, e_{16} \oplus e_1\}$, where e_i is the 16-bit unit vector in position i in Figure 4.8, Figure 4.9 and Figure 4.10, since the number of curves is very huge, 2^{16}. Namely, the Hamming Weight of first 16 input differences is 1, the Hamming Weight of last 16 input differences is 2. Moreover, SAC curves are computed for $i \in F_2^n - \{0\}$. In Figure 4.8, Figure 4.9 and Figure 4.10 different colors correspond to different input differences.

4.4.2 Results for the FI Function of MISTY1

Figure 4.8 shows the normalized distance to SAC values of MISTY1's FI function, explained in section 2.1.1, versus j. The highest value is obtained as $|D|_{\max} = 1$ for eleventh bit position of avalanche vector and the input difference e_9. It is an interesting observation that there is not any maximum value, $|D_j^i|_{\max} = 1$, for $1 \le j \le 7$. Moreover, $|D_j^i| = 0$ for $1 \le j \le 5$ and $i \in \{e_1, \ldots, e_{16}, e_1 \oplus e_2, e_2 \oplus e_3, \ldots, e_{15} \oplus e_{16}, e_{16} \oplus e_1\}$. Although most values are equal to 0, the FI function does not satisfy SAC, because of the nonzero values of $|D_j^i|$.

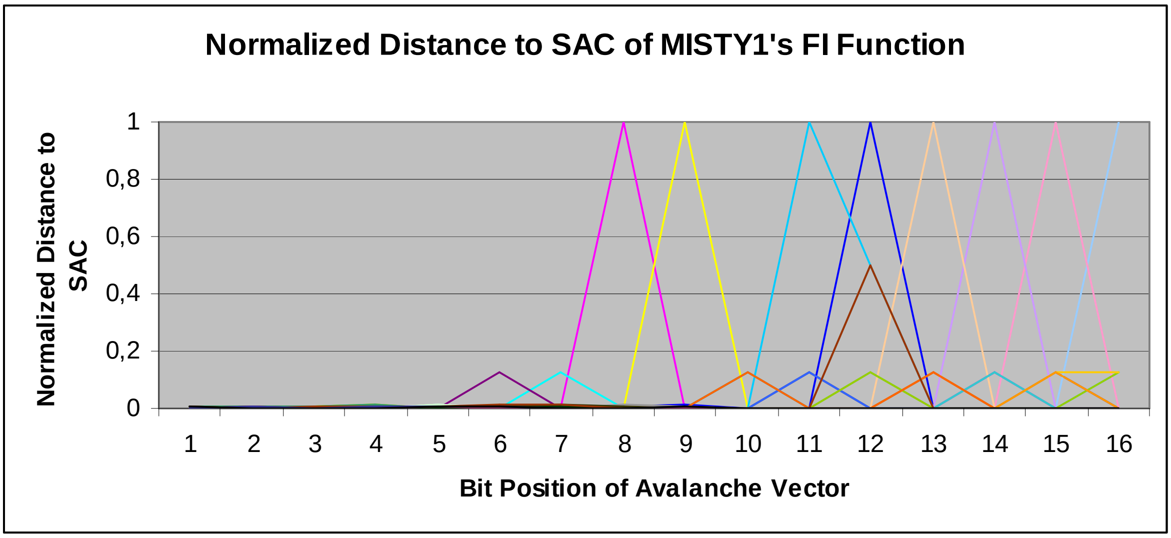

Figure 4.8 Normalized Distance to SAC of MISTY1's FI Function

4.4.3 Results for the FI Function of KASUMI

The normalized distance to SAC values of KASUMI's FI function, explained in section 2.2.1, are depicted in Figure 4.9. The maximum of normalized distance to SAC is obtained for the input difference e_6 with $\left|D_j^i\right|_{\max} = 0.34$ and $j = 12$. All values are closer to zero than those of MISTY1. On the other hand, similar to previous case, normalized distance to SAC values are very close to 0 for $1 \le j \le 5$.

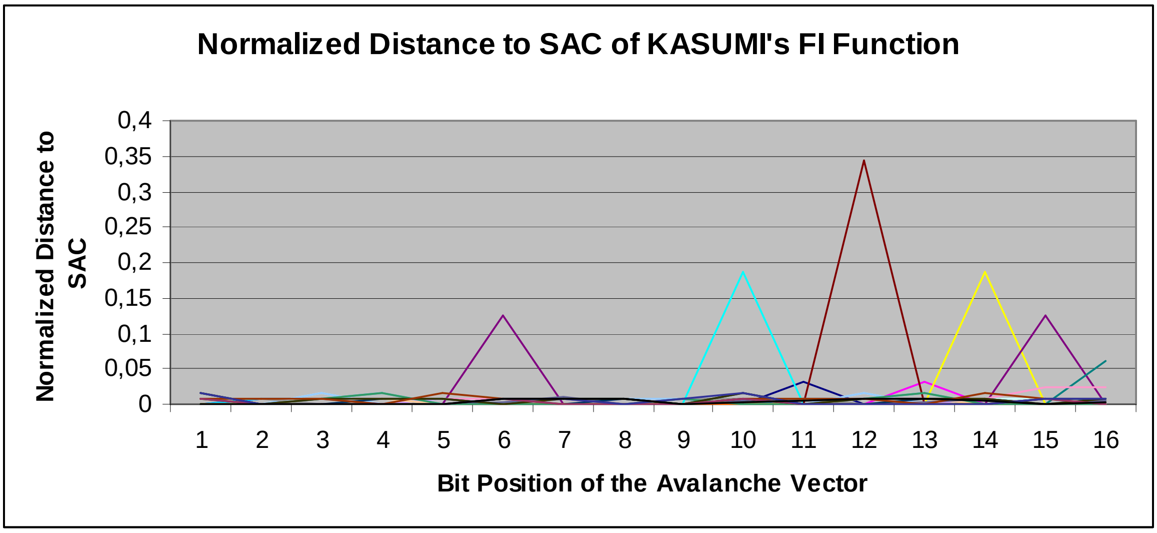

Figure 4.9 Normalized Distance to SAC of KASUMI's FI Function

4.4.4 Results for the FI Function of KASUMI-R

As the previous two figures, Figure 4.10 gives the normalized distance to SAC values of KASUMI-R's FI function explained 4.4.1. The highest value is obtained as $\left|D_j^i\right|_{\max} = 0.027$ for $j = 8$ and the input difference $e_3 \oplus e_4$. Although FI function with RIJNDAEL s-box does not satisfy the strict avalanche criteria, its normalized distance to SAC values are very close to 0 which is highly satisfactory. In FI function with RIJNDAEL s-box, all values are in a very small gap. As an interesting observation, we note that the proportion of zeros of $\left|D_j^i\right|$ is only 10 over 512 as opposed to other FI functions.

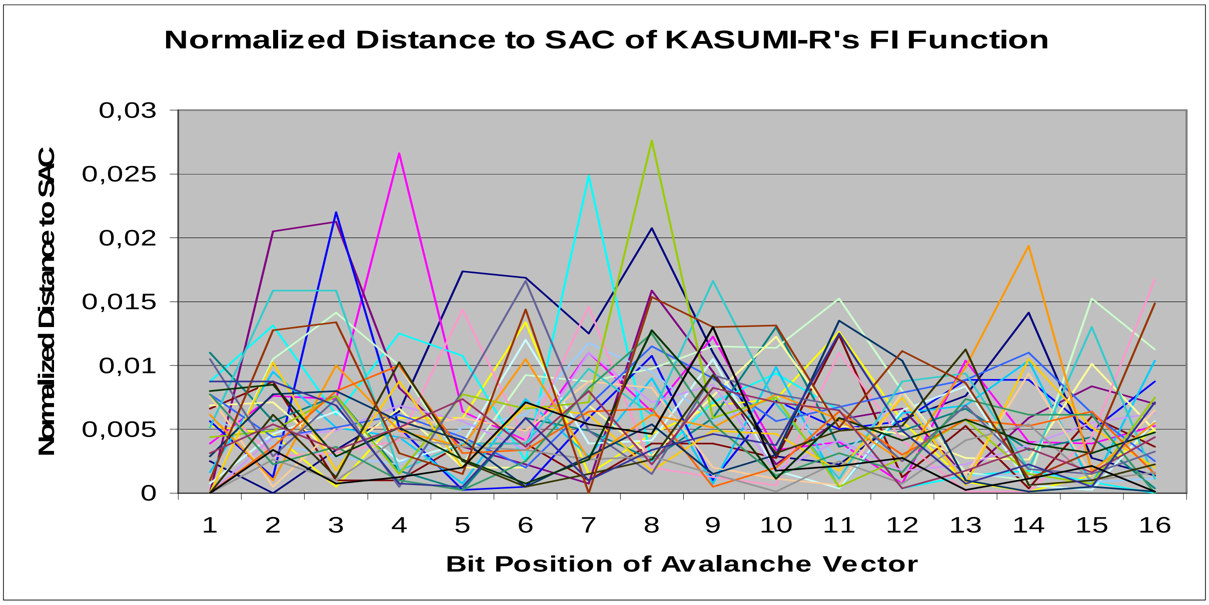

Figure 4.10 Normalized Distance to SAC of KASUMI-R’s FI Function

4.4.5 Comparison of FI Functions of MISTY, KASUMI and KASUMI with RIJNDAEL S-box

In Figure 4.11, we sketch the SAC performances of the three FI functions of MISTY1, KASUMI and KASUMI-R, in terms of the normalized distance $\left|D_j^i\right|$ versus $j = 1,\ldots,16$ for the input difference set $i \in \{e_1,\ldots,e_{16}, e_1 \oplus e_2, e_2 \oplus e_3,\ldots,e_{15} \oplus e_{16}, e_{16} \oplus e_1\}$. It is observed that the FI function of KASUMI-R has the best performance and the corresponding $\left|D_j^i\right|$ values are very small as compared to the FI functions of MISTY1 and KASUMI. To make these small values more visible, we zoom at the vertical scale of Figure 4.11 and obtain Figure 4.12, which shows the KASUMI-R performance in more detail. In Figure 4.11 and Figure 4.12 each color corresponds to a cipher, i.e., blue is for MISTY1, red is for KASUMI and yellow is for KASUMI-R. It should be noticed that FI function of MISTY1 and KASUMI get their maximum normalized distance to SAC value for input difference of weight one as opposed to KASUMI-R’s FI function. It is an interesting observation that all FI functions get their maximum values for $j > 7$, where j is the bit position of the avalanche vector.

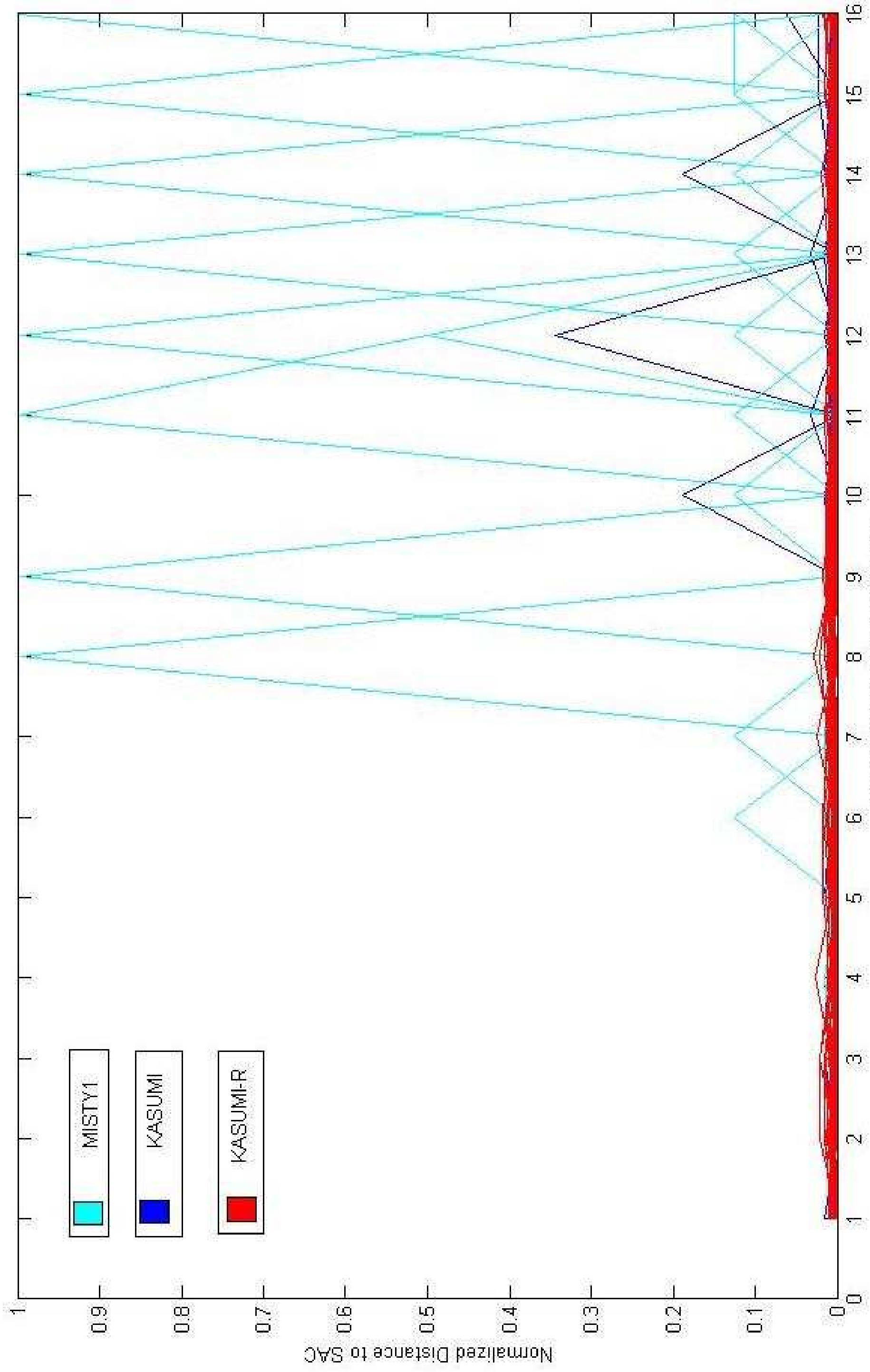

Figure 4.11 Normalized Distance to SAC of FI Functions with All Values

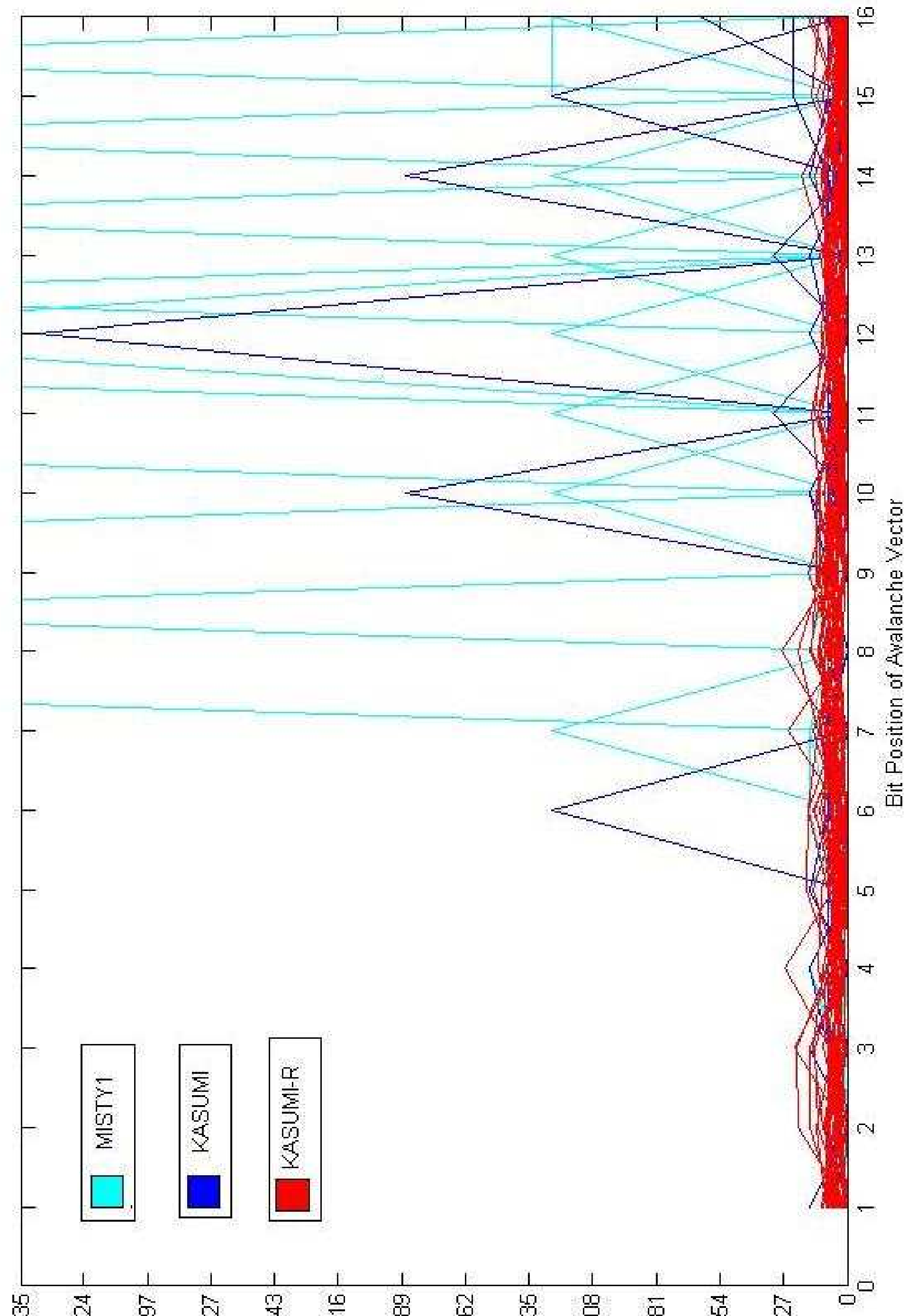

Figure 4.12 Normalized Distance to SAC of FI Function

As a conclusion, normalized distance to SAC values show that KASUMI-R seems to be the most random. We finally find the SAC parameters for all input differences, $F_2^{16}-\{0\}$, and see that the highest normalized distance to SAC values are 1, 0.34 and 0.027 for the FI functions of MISTY1, KASUMI and KASUMI-R, respectively. As an interesting observation that when the input difference set $i \in \{e_1,...,e_{16},e_1 \oplus e_2, e_2 \oplus e_3,...,e_{15} \oplus e_{16}, e_{16} \oplus e_1\}$ is used, we obtain the maximum values for the three FI functions. Table 4.17 summarizes maximum values of the normalized distance SAC for FI functions for all input differences.

Table 4.17. Normalized Distance to SAC for FI Functions

FI Function of	$\left\|D_j^i\right\|_{\max}$	**Corresponding** i
MISTY	1	256
KASUMI	0.34	32
KASUMI-R	0.027	222

The maximum values, given in Table 4.17, can also be interpreted by using $P\{FI(x) \oplus FI(x \oplus i) = 0\} = \frac{1}{2} + \frac{|D_i^j|_{\max}}{2}$ as follows:

Table 4.18. Probabilities for FI Functions in terms of SAC

FI Function of	**Probability Function**
MISTY	$P\{FI(x) \oplus FI(x \oplus e_9) = 0\} = 1$
KASUMI	$P\{FI(x) \oplus FI(x \oplus e_6) = 0\} = 0.67$
KASUMI-R	$P\{FI(x) \oplus FI(x \oplus e_3 \oplus e_4) = 0\} = 0.5135$

CHAPTER 5

RANDOMNESS CRITERIA FOR BLOCK CIPHERS

In this chapter, Avalanche Weight Distribution (AWD) and some statistical randomness tests are explained in detail to examine diffusion, confusion and randomness properties of overall cipher.

5.1 Avalanche Weight Distribution

Avalanche Weight Distribution (AWD) is defined in [1] as a simple criterion for fast and rough analysis of the diffusion and confusion properties mentioned by Shannon. This criterion examines whether for quite similar plaintext pairs (P_1, P_2), histograms of the Hamming weight of the avalanche vectors are completely random. For a well diffused block cipher of blocklength n AWD curves corresponding to all possible pairs of similar inputs should be binomially distributed around $n/2$. In order to give an idea about what is expected from AWD of a random block cipher, Binomial distribution is sketched for $n = 64$ in Figure 5.1.

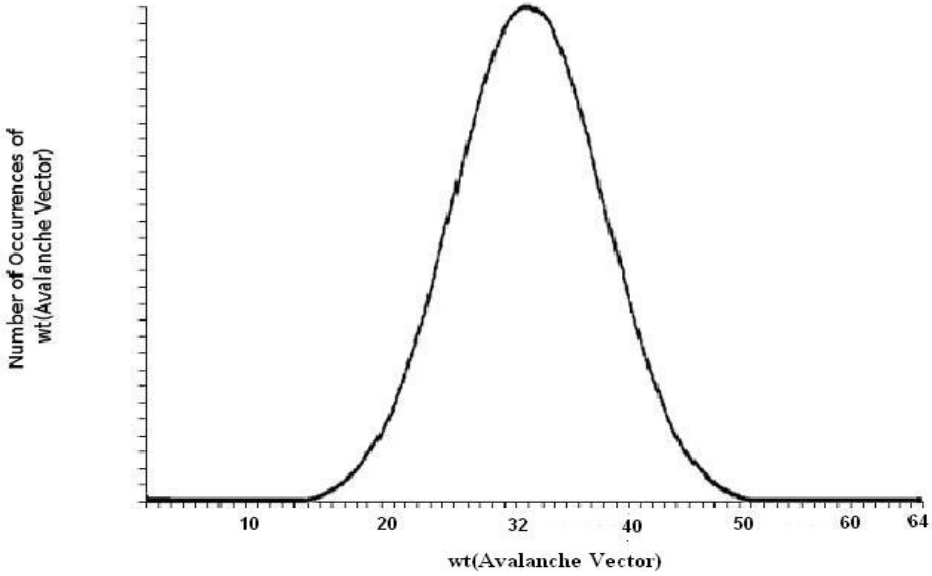

Figure 5.1 Ideal AWD Curve

The following test procedure is used to find the randomness of the r-th round output of an R-round cipher which maps n-bits to n-bits. The avalanche weight distribution vector with the k-th element AWD[k] denotes the number of avalanche vectors of weight k.

Step 0 : Choose $N = 30000$ random input vectors P.

Step 1 : Set AWD[k]=0, for $k \in \{1,...,64\}$.Choose r and i such that $1 \leq r \leq R$ and $1 \leq i \leq n$. Do the followings for each input vector P:

Step 2 : Calling e_i, the n-bit unit vector having a 1 at position i, compute $P_{e_i} = P \oplus e_i$. P and P_{e_i}, differ only in bit i.

Step 3 : Submit P and P_{e_i} r-rounds of the cipher, call the r^{th} round outputs of $f(P)$ and $f(P_{e_i})$.

Step 4 : Find the Hamming weight of k the avalanche vector $f(P) \oplus f(P_{e_i})$.

Step 5 : Increment the value of the k^{th} element of the avalanche weight distribution vector, i.e., AWD[k] = AWD[k] + 1.

Step 6 : Return Step 2 until all input vectors are exhausted.

Using the above algorithm n times, one obtains n different AWD curves for n different input difference e_i vectors. Similarly, letting $r = 1,2,...,8$, one can compute the randomness of the cipher for different number of rounds. Moreover, one can test the randomness of the round output increases as r is augmented. By using the test procedure explained above, we investigate avalanche weight distribution of 64 different input difference e_i vectors for $1 \leq i \leq 64$ overall test of MISTY, KASUMI and KASUMI-R ciphers. Figure 5.2, Figure 5.3 and Figure 5.4 demonstrate eight different AWD curves, each corresponds to different color and rounds, for the input difference e_1.

5.1.1 AWD Test Results for MISTY1

Figure 5.2 shows the AWD curves of MISTY1 for the input difference e_1 at the end of different rounds of the cipher. We observe that MISTY1 satisfies the AWD criterion at the end of the second round. After rounds of encryption all the avalanche vectors have very stable Hamming weights, gathered around a mean of 32. The AWD curves almost imitate each other independent of round number except the first round. In the first round the AWD curve is similar to a binomial distribution a mean value of 24.

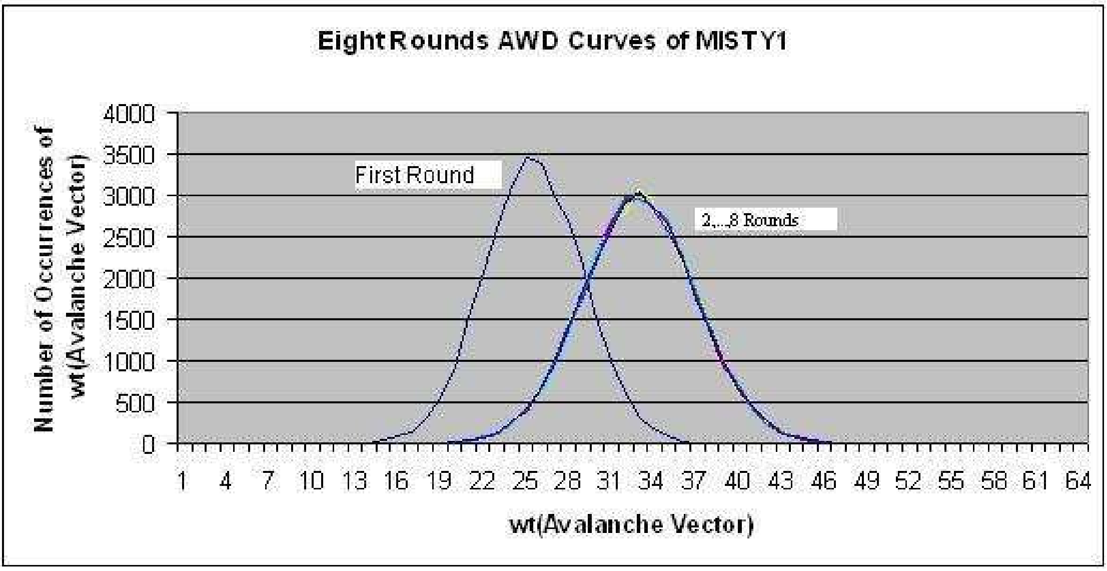

Figure 5.2 AWD Curves of MISTY1 for e_1 for all Rounds

5.1.2 AWD Test Results for KASUMI

Eight different AWD curves of KASUMI corresponding to round number r for the input difference e_1 are depicted in Figure 5.3. After augmenting the number of rounds to 2, we observe that KASUMI satisfies the AWD criterion. Moreover, when the round number is more than 1, all AWD curves are very similar to each other and very close to the binomial shape. On the other hand, in the first round, the Hamming weight of the avalanche vectors is aggregated around 19.

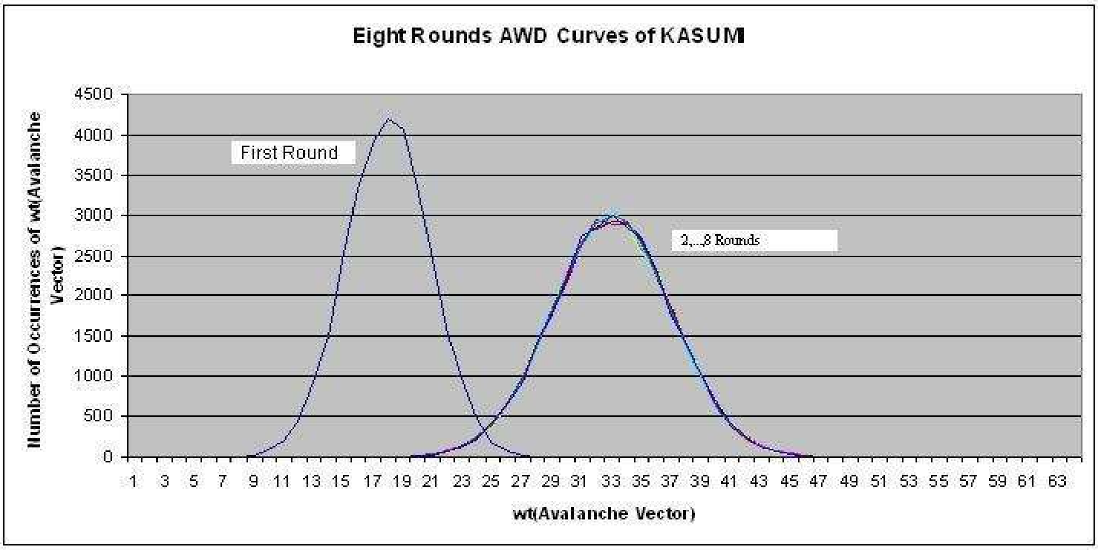

Figure 5.3 AWD Curves of KASUMI for e_1 for all Rounds

5.1.3 AWD Test Results for KASUMI-R

Figure 5.4 demonstrates AWD curves of KASUMI-R for the input difference e_1 for the eight different rounds. In the first round it resembles a binomial distribution a mean value of 18. Increasing the number of rounds to 2 causes satisfactory AWD curve. Immediately after the second round the Hamming weight of the avalanche vectors are gathered around a mean of 32. Moreover, there is not much difference of AWD curves after the second round.

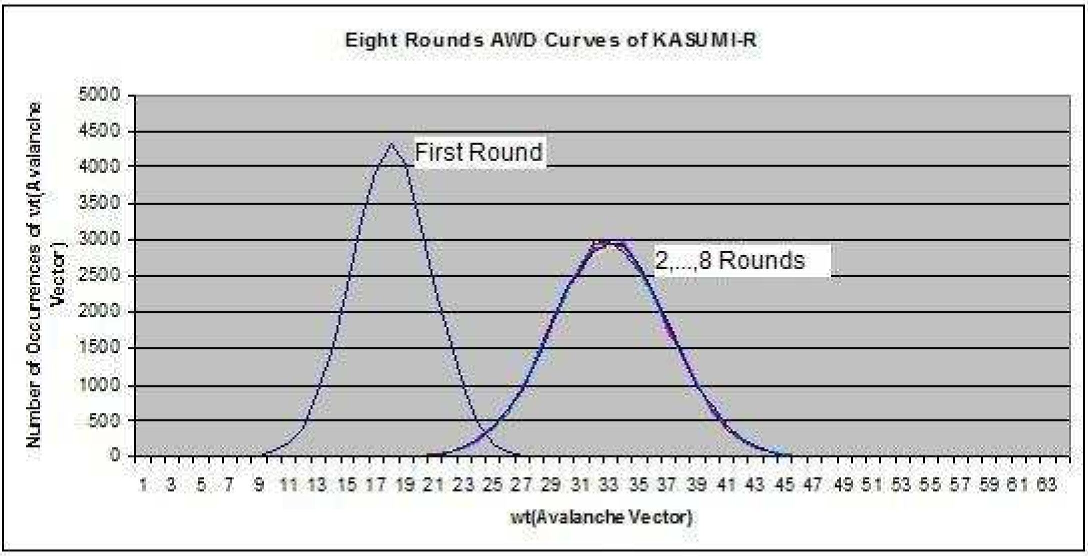

Figure 5.4 AWD Curves of KASUMI-R for e_1 for all Rounds

A normalized measure of closeness is defined as in [1] between the evaluated AWD curves and the binomial distribution, subtracting the sum of normalized error magnitudes from unity, as:

$$R_r^{e_i} = 1 - \frac{1}{2N}\sum_{k=0}^{n} | AWD[k] - \frac{N}{2^n}\binom{n}{k} | \qquad (5.1)$$

where n is the length of the ciphertext and N is the number of plaintexts and r is the number of rounds.

5.1.4 Comparison of MISTY1, KASUMI and KASUMI-R in view of AWD

When we compare the AWD curves results, it can be concluded that all ciphers show similar behaviors input differences for e_i the 64-bit unit vector in position i. The difference between them is the Hamming weights of avalanche vectors for the first round.

Normalized measure of closeness between the evaluated AWD curves for MISTY1, KASUMI, KASUMI-R and the ideal Binomial distribution (see 5.1) are given in Table 5.1 for the input difference e_1 for all rounds. It is observed from Table 5.1 that almost all rounds except the first round of MISTY1, KASUMI and KASUMI-R resemble the ideal binomial curve of mean value 32 more than 98%. At the end of the first round MISTY1 has the best result in terms of resemblance percentages, since AWD curve resembles a binomial distribution a mean value of 24 which is close to 32 than others. As a conclusion evaluated curves of KASUMI are much closer to the binomial distribution of mean value 32. The reason why all ciphers fail for the first round may be the Feistel structure. For input differences of weight one, we obtain similar AWD curves for the three ciphers.

Table 5.1. Resemblance Percentages $R_r^{e_1}$ for MISTY1, KASUMI and KASUMI-R

Round *r*	**MISTY1**	**KASUMI**	**KASUMI-R**
1	0.3893	0.038	0.039
2	0.9808	0.9865	0.9831
3	0.9815	0.9854	0.9901
4	0.9833	0.9890	0.9867
5	0.9827	0.9836	0.9853
6	0.9858	0.9841	0.9888
7	0.9849	0.9901	0.9854
8	0.9860	0.9923	0.9863

5.2 Randomness Testing

Being randomness is one of the important criteria to evaluate block ciphers. The output of the block cipher should not give any hint that enables to distinguish it from a truly random sequence. Therefore, statistical tests are used to verify that whether a sequence is completely random. Two statistical testing for randomness are described in this part.

5.2.1 Frequency (Monobit) Test [15]

Frequency test is based on the weight of the sequence. The purpose of this test is to determine whether the number of 0’s and 1’s in a sequence are approximately the same, as would be expected for a random sequence.

5.2.1.1 Test Description

Step 1 : The zeroes of the sequence $S = (S_1, S_2, ..., S_n)$ are converted to values -1, $S_i = 2S_i - 1$, where $1 \le i \le n$. The length of sequence should be at least 100.

Step 2 : Compute $|S| = \sum_{i=1}^{n} S_i$.

Step 3 : Compute the test statistic $S_{obs} = \frac{|S|}{\sqrt{n}}$.

Step 4 : Compute $p-value = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right)$, where erfc is the complementary error function as $2\cdot\int_z^{\infty}\frac{1}{\sqrt{\pi}}e^{-x^2}dx = 2\left(1-normcdf(z,0,\frac{1}{\sqrt{2}})\right)$.

Step 5 : If the computed p-value is greater than 0.01 (decision rule at the 1% level), then conclude that the sequence is random.

5.2.2 Frequency Test within a Block [15]

The purpose of this test is to determine whether the frequency of ones in an M-bit block is approximately $M/2$, as would be expected under an assumption of randomness. Frequency test within block is based on the weight of the sequence.

5.2.2.1 Test Description

Step 1 : Partition the input sequence into $N = \left\lfloor \frac{n}{M} \right\rfloor$ non-overlapping blocks. M should be greater than 20. Sequences are defined as $S^i = (S_1^i, S_2^i, ..., S_M^i)$ for $1 \le i \le N$. Note that if $N = 1$, then this test is the same as frequency (monobit test).

Step 2 : Compute the proportion of ones, a_i, in each M-bit block, $a_i = \frac{\sum_{j=1}^{M} S_j^i}{M}$ for $1 \le i \le N$.

Step 3 : Compute the test statistic $\chi^2(obs) = 4M\sum_{i=1}^{N}(a_i - 1/2)^2$.

Step 4 : Compute $p-value = igamc\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right)$, where igamc is the incomplete gamma function as $\int_{\chi^2(obs)}^{\infty}\frac{e^{-u}\cdot u^{\frac{N}{2}-1}}{\Gamma(N/2)}du = 1-gamcdf\left(\frac{\chi^2(obs)}{2}, \frac{N}{2}, 1\right)$.

Step 5 : If the computed p-value is greater than 0.01 (decision rule at the 1% level), then conclude that the sequence is random.

5.3 Test Results for Randomness

Randomness tests are performed using the strategy explained in [18] and for each data set separately to check whether the output is random :

i) Plaintext Avalanche, Plaintext-Ciphertext Correlation and Low Density Plaintext data types are analyzed.

1. Plaintext Avalanche

100 binary sequences are analyzed to examine the sensitivity of individual algorithms to changes in the plaintext. The 10000 sequences are parsed from a string constructed as follows : given 10000 random 64-bit plaintext blocks and 128-bit key of all zeroes, 6400 derived blocks are concatenated. Then, 409600 = 64x64x100 is obtained for each sequence. Each derived block is based on the XOR of the ciphertext formed using the fixed 128-bit key and the random plaintext, and the ciphertext formed using the fixed 128-bit key and the perturbed random plaintext with the i^{th} bit changed, for $1 \leq i \leq 64$. Plaintext avalanche can be explained mathematically as follows:

$\bigcup_{i=1}^{100} B_i = \bigcup_{j=1}^{10000} \left(\bigcup_{l=1}^{64} f(P_j, K) \oplus f(P_j \oplus e_l, K) \right)$, where $\cup$ is used to concatenate the vectors, each B_i is 409600-bit vector and $f(P,K)$ is encryption function with input P, key K, e_l is a 64-bit unit vector in position l.

2. Plaintext/Ciphertext Correlation

100 binary sequences (409600 bits per sequence) are analyzed to examine the sensitivity of individual algorithms to changes in the plaintext. Given a random 128-bit key and 10000 random plaintext blocks, a binary sequence is constructed concatenating 10000 derived blocks (where a derived block is the result of applying the XOR operator on the plaintext block and its corresponding ciphertext block computed). Plaintext/Ciphertext correlation can be explained mathematically as follows:

$\bigcup_{i=1}^{100} B_i = \bigcup_{j=1}^{64}\left(\bigcup_{l=1}^{10000} P_l \oplus f(P_l, K_j) \right)$, where $\cup$ is used to concatenate the vectors, each B_i is 409600-bit vector and $f(P,K)$ is encryption function with input P, key K.

3. Low Density Plaintext

Each data set created based on low density plaintext blocks consisted of 11 sequences. Ciphertext blocks is calculated using plaintext blocks consisting of a single one and 63 zeros, the one appearing in each of the 64 bit positions of the plaintext block or none. The other plaintext blocks had two ones and 62 zeros, the ones appearing in each combination of two positions of the plaintext block. Totally, 64+2015=2079 different plaintexts are generated. Low density plaintext can be explained mathematically as follows:

$\bigcup_{i=1}^{11} B_i = \bigcup_{j=1}^{64}\left(\bigcup_{l=1}^{2079} f(P_l, K_j) \right)$, where $\cup$ is used to concatenate the vectors, each B_i is 12096-bit vector and $f(P,K)$ is encryption function with input P, key K, e_l is a 64-bit unit vector in position l. 2079 input vector are given such that $l = 1 \Rightarrow wt(P_l) = 0$, $2 \le l \le 65 \Rightarrow wt(P_l) = 1$, where P_l is a 64-bit unit vector in position l, $66 \le l \le 2079 \Rightarrow wt(P_l) = 2$

ii) Input parameters are fixed for plaintext avalanche and plaintext/ciphertext correlation. These parameters are set at 409600 bits for frequency test and 4096 bits for block frequency test, 100 binary sequences for frequency test and 10000 binary sequences for block frequency test and 0.01, sequence length, sample size and significance level (p-value), respectively. For low density plaintext parameters are set 12096 bits for frequency test and 4096 for block frequency test, 11 binary sequences for frequency test and 33 binary sequences for block frequency test and 0.01, sequence length, sample size and significance level, respectively.

iii) For each p-value, a success/failure assessment is made on whether or not it exceeded or fell below the pre-selected significance level.

5.3.1 Frequency within a Block Test Results for MISTY1, KASUMI and KASUMI-R

In this part length of the sequences is fixed to 64x64 = 4096 for plaintext avalanche and plaintext/ciphertext correlation data. Then, 10000 binary sequences are obtained to test. However, we give only 100 sequences in Figure 5.5 and Figure 5.6. For low density plaintext, length of sequences is again fixed to 64x64 = 4096 to interpret results with the previous data. Then, 133056/4096, 33 binary sequences are obtained to test. Note that the last sequence has only 1984 elements. Figure 5.5, Figure 5.6 and Figure 5.7, which shows the percentages of 1's, contains only three rounds first, second and last rounds, of the three ciphers for Test 5.2.2 since all ciphers get randomness at the end of the second round. p-values are calculated by using

$$p-value = igamc\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) \text{ (see 5.2.2.1).}$$

In Figure 5.5, percentages of 1's are away from the desired result for the first round for plaintext avalanche data type. By the second round all ciphers possess the randomness in terms of weight of the sequence and this goes until the last round. p-values are greater than 0.01 immediately after the second round of all ciphers. Their distribution is quite similar to each other for the second and last round. However, in the first round KASUMI has the worst case since percentage of 1's is around 25%.

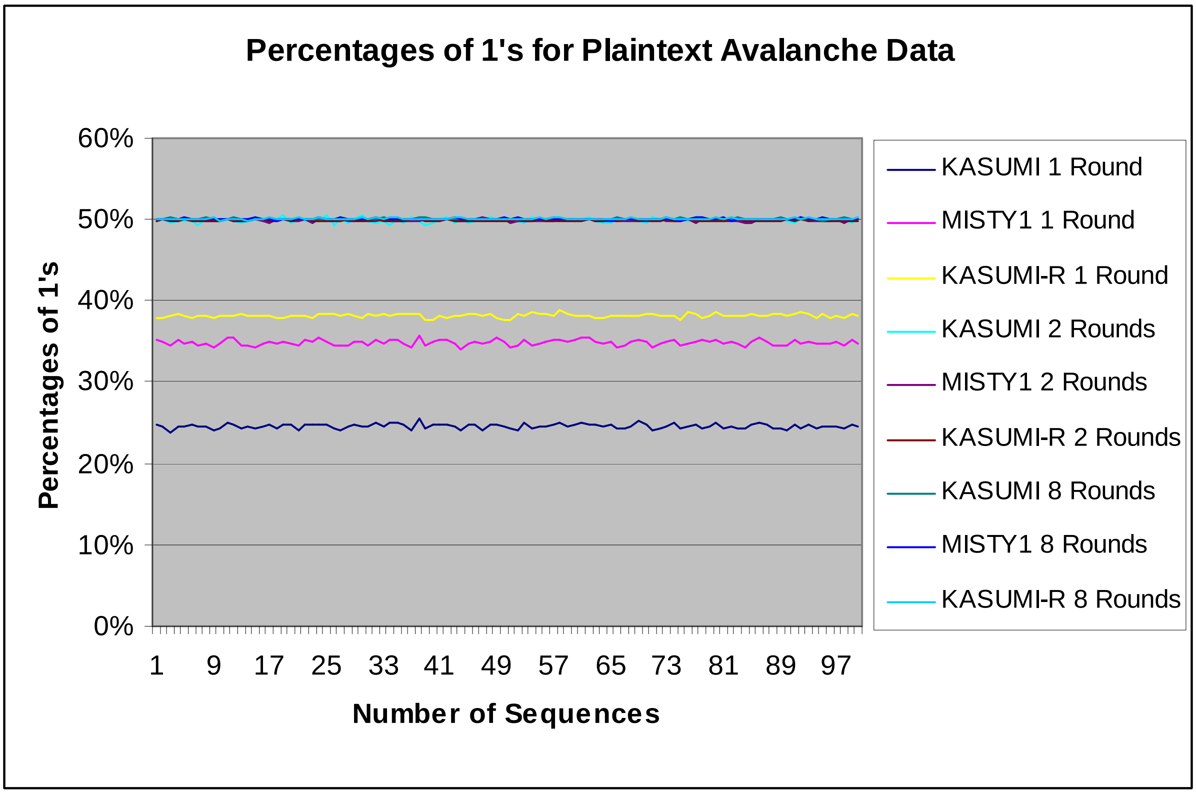

Figure 5.5 Percentages of 1’s for Plaintext Avalanche

Percentages of 1’s for plaintext/ciphertext correlation for the three ciphers for the first, second and last rounds are depicted in Figure 5.6. Similar to previous case, ciphers have poorly random function for the first. By the second round all ciphers have equally likely zeroes and ones. p-values are very close for the second round of all ciphers. There is no significant difference between the second and last rounds of the three ciphers. However, in the first round KASUMI-R has the best case like the previous case.

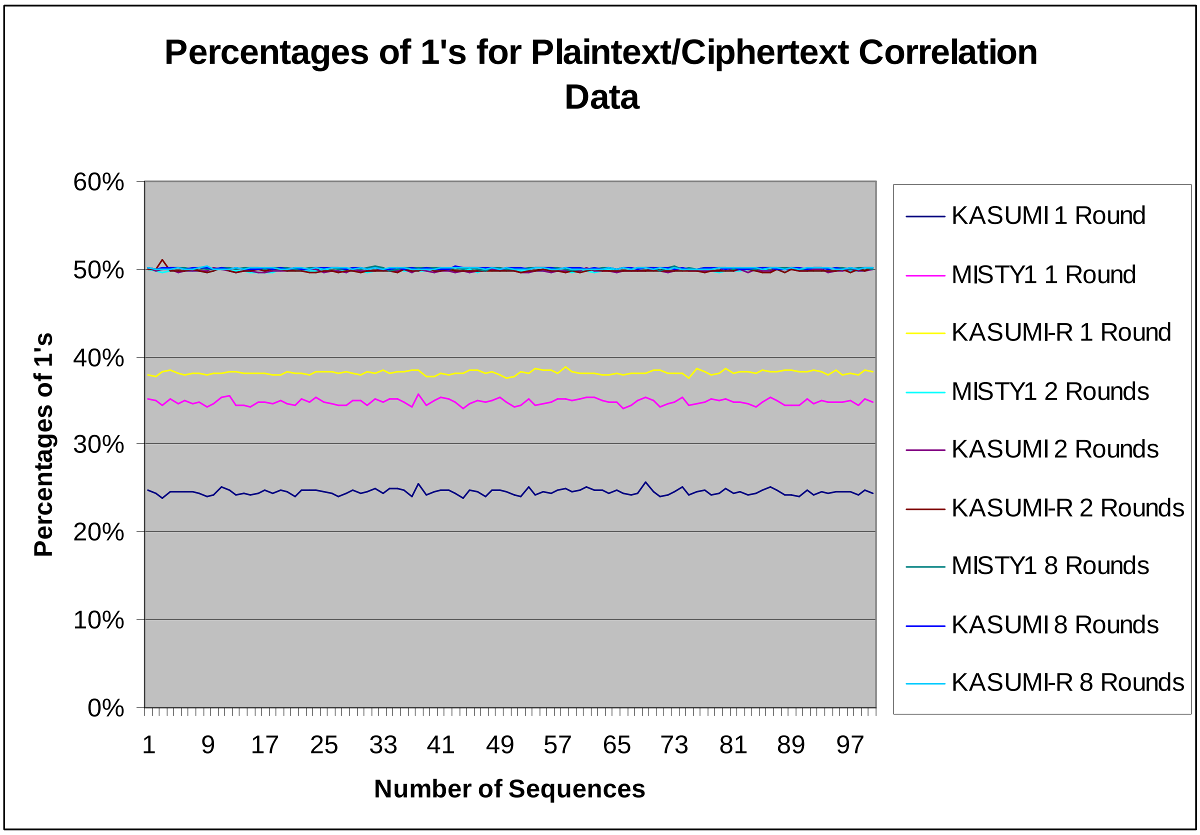

Figure 5.6 Percentages of 1’s for Plaintext/Ciphertext Correlation

Figure 5.7 shows percentages of 1’s for low density data for the ciphers for the first, second and last rounds. Outputs of the all ciphers have too many zeroes at the end of the first round. Therefore, satisfactory randomness is not achieved for the first round for low density plaintext data type. By the second there is no significant deviation from 50%. p-values are greater than 0.01 immediately after the second round of all ciphers. Their distribution is quite similar to each other for second and last round except the first round.

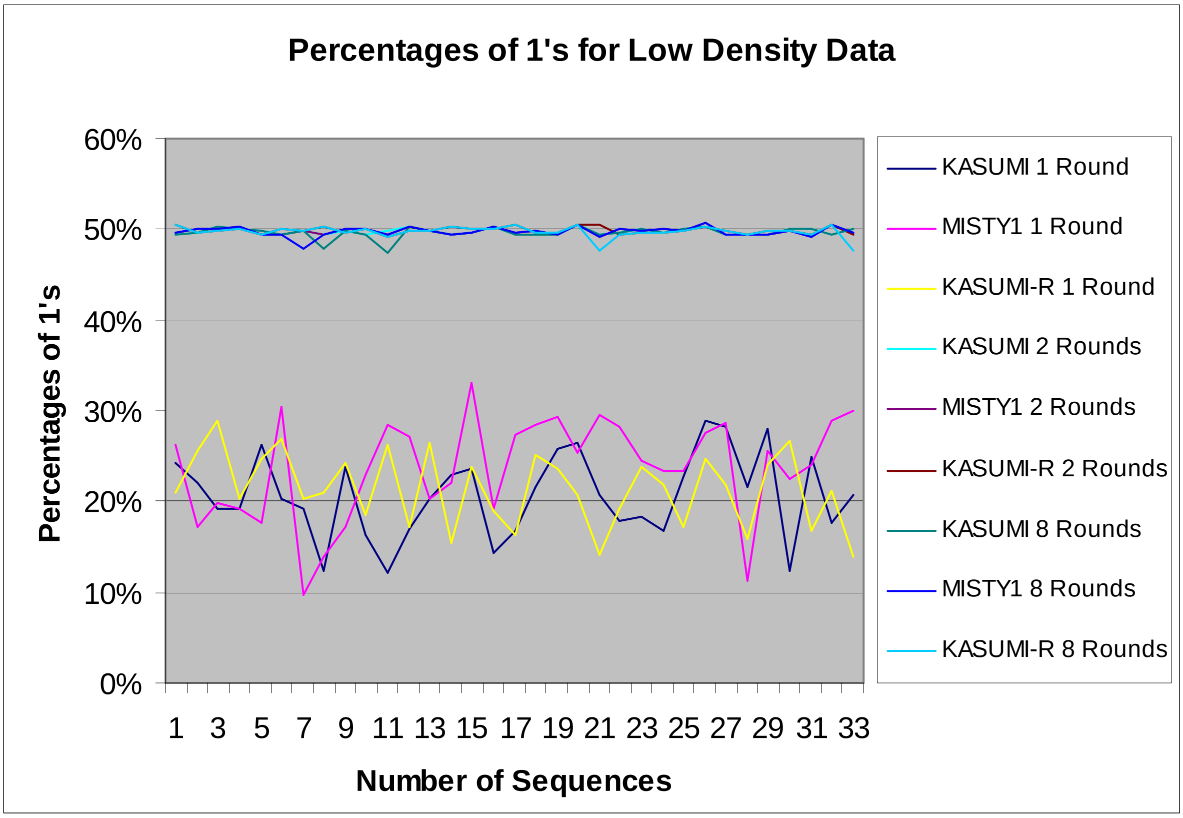

Figure 5.7 Percentages of 1's for Low Density Plaintext

By concatenating 100 sequences for plaintext avalanche, plaintext/ciphertext correlation, we obtain 100 sequences to implement frequency test. Length of these sequences is 64x64x100=409600. Similarly, after concatenating 33 sequences for low density plaintext, we divide this sequence, 64x64+1984=133056, into 11 equal parts. Length of the each sequence is 12096. Then, since satisfactory test results are obtained for frequency test within a block after the second round, we can say that all ciphers satisfy frequency test.

5.4 Comparison of Avalanche Weight Distribution and Randomness Test Results

According to avalanche weight distribution, monobit test and frequency test within a block test, MISTY1, KASUMI and KASUMI-R reach the randomness at the end of the second round.

CHAPTER 6

CONCLUSION

In this thesis, KASUMI, the standard algorithm for the 3rd Generation GSM, MISTY1 and KASUMI–R, defined in Section 4.4.1, are studied in order to see their performance in terms of the satisfaction of some cryptographic test criteria appearing in the literature. Our investigation is mainly about the s-boxes and FI functions of these ciphers. Table 6.1 summarizes obtained test results of the five s-boxes.

Table 6.1 Summary of Test Results for S-boxes

Test	$\max_{a,c\in F_2^n} \lvert LAT(a,c) \rvert$	**Nonlinearity**	**Differential Uniformity**	$\lvert D_i^j \rvert_{\max}$
S-box of				
MISTY1 7x7	8	56	2	0.125
KASUMI 7x7	8	56	2	0.125
RIJNDAEL 8x8	16	112	4	0.125
MISTY1 9x9	16	240	2	1
KASUMI 9x9	16	240	2	1

S-boxes of MISTY1 7x7, KASUMI 7x7 have essentially the same cryptographic properties in terms of SAC, LAT and XOR table distributions. Considering partial LAT and XOR table distributions, 9x9 s-box of KASUMI has zeros at the same places with those of LAT and XOR table distributions. There are some differences between RIJNDAEL's s-box and other s-boxes in terms of the number of different elements in LAT and XOR table. RIJNDAEL's 8x8 s-box has 17 different LAT values and 3 different XOR values, whereas MISTY1's and KASUMI's s-boxes have only 3 different values and 2 different values in their LAT and XOR table, respectively. Moreover, distribution of LAT elements is equal for the s-boxes of MISTY1 and KASUMI. Similarly, distribution of XOR table elements is also the same. 7x7 and 8x8 s-boxes satisfy SAC within very small deviations. On the other hand, 9x9 s-boxes yield the most undesired value of $\lvert D \rvert_{\max}$ =1 for the deviation from SAC.

The FI function of KASUMI-R has the best performance and the corresponding $\left|D_j^i\right|$ values are very small as compared to the FI functions of MISTY1 and KASUMI. In addition, we finally find the SAC parameters for all input differences, $F_2^{16}-\{0\}$, and see that the highest normalized distance to SAC values are 1, 0.34 and 0.027 for the FI functions of MISTY1, KASUMI and KASUMI-R, respectively.

Overall performances of MISTY1, KASUMI and KASUMI-R according to the AWD criterion show that AWD curves of 64-bit vectors resemble a binomial distribution around a mean value of 32 at the end of the second round. They all yield very similar curves independent of the number of rounds except for the first round. In the first round it resembles a binomial distribution with mean values of 24, 19 and 18 for MISTY1, KASUMI and KASUMI with RIJNDAEL s-boxes, respectively.

The two core tests of NIST, the monobit test and the frequency test within a block are implemented for the plaintext avalanche, plaintext/ciphertext correlation and low density plaintext kinds of data in order to calculate the randomness. The test results are similar for all data types and tests, i.e., at the end of the second round, the number of 1's seems quite random for all data types.

Our observations on the MISTY1, KASUMI and KASUMI-R do not indicate any hint that one of these is superior to the others. The differences between FI functions of the three ciphers and their s-boxes seem to have no observable affect on the overall cryptographic strength of these ciphers.

REFERENCES

[1] Ekrem Aras, Analysis of Security Criteria for Block Ciphers, M.S. Thesis, Middle East Technical University, Turkey, 1999.

[2] E. Biham and A. Shamir, Differential Cryptanalysis of DES-Like Cryptosystems, Journal of Cryptology, Volume: 4, pages 3-72, 1991.

[3] A. Biryukov and A. Shamir, Real Time Cryptanalysis of the Alleged A5/1 on a PC, FSE'00, 2000.

[4] M. Briceno, I. Goldberg and D. Wagner, A Pedagogical Implementation of the GSM A5/1 ans A5/2 Voice Privacy Encryption Algorithms, 1999.

[5] J. Daemen and V. Rijmen, The Design of Rijndael, Springer, 2002.

[6] H. Dobbertin, Almost Perfect Nonlinear Power Functions on GF(2^n): The Welch Case, IEEE Transactions on Information Theory, Vol. 45, No.4, 1999.

[7] Orr Dunkelman, Comparing MISTY and KASUMI, NES/DOC/TEX/WP5/029/a, 2002.

[8] H. Feistel, Cryptography and Computer Privacy, Scientific American, Volume 228, No: 5, pp.15-23, 1973.

[9] H. Heys, A Tutorial on Linear and Differential Cryptanalysis, Cryptologia, Vol. XXVI, No.3, pp.189-221, 2002.

[10] F. Hillebrand, GSM and UMTS: The Creation of Global Mobile Communication, Springer, 2007.

[11] J.B. Kam and G.I. Davida, Structured Design of Substitution-Permutation Encryption Networks, IEEE Transactions on Computers, Volume C-28, No:10, pp.747-753, 1979.

[12] M. Matsui, Linear Cryptanalysis Method for DES Cipher, Eurocrypt'93, Lectures Notes in Computer Science, No. 765, Springer Verlag, pp. 386-397, 1994.

[13] M. Matsui, New Block Encryption MISTY, FSE 4th International Workshop, Volume 1267 of Lecture Notes in Computer Science, pages 54-68, Springer Verlag, 1997.

[14] K. Nyberg, Differentially Uniform Mappings for Cryptography, EUROCRYPT'93, Lecture Notes in Computer Science, Vol. 765, Springer Verlag, pp. 55–64, 1994.

[15] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, A Statistical Test Suite for Random and Pseduorandom Number Generators for Cryptographic Applications, NIST Special Publication 800-22, 2001.

[16] J. Scourias, A Brief Overview of GSM, University of Waterloo, 1994.

[17] Claude E. Shannon, Communication Theory of Secrecy Systems, Bell System Technical Journal, vol.28-4, page 656--715, 1949.

[18] J. Soto and L. Bassham, Randomness Testing of the Advanced Encryption Standard Finalist Candidates, NIST 2000.

[19] A. Webster and S. Tavares, On the Design of S-boxes, Advances in Cryptology, Proc. Eurocrypt'85, Springer Verlag, pp.523-534, 1986.

[20] Melek D. Yücel, Alternative Nonlinearity Criteria for Boolean Functions, Departmental Memorandum, No:2001-1, 2001.

[21] 3G TR 33.901 3G Security Criteria for Cryptographic Algorithm Design Process V 3.0.0, 1999.

[22] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security, Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification, V.3.1.1, 2001.

[23] CRYPTREC, Cryptography Research and Evaluation Committee, Homepage online. http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html [Accessed 08 February 2008]

[24] ETSI/SAGE, KASUMI Specification, Part of the Specification of the 3GPP Confidentiality and Integrity Algorithms, 1999.

[25] NESSIE, New European Schemes for Signature, Integrity and Encryption. Homepage on-line. http://www.cryptonessie.org [Accessed 08 February 2008]

[26] http://www.3gpp.org/about/about.htm [Accessed 08 February 2008]

[27] http://cryptodox.com/A5/2 [Accessed 08 February 2008]