Analysis of Linear Relationships in Block Ciphers

by

Muhammad Reza Z’aba

Bachelor of Science (Computer) (Universiti Teknologi Malaysia) – 2004

Thesis submitted in accordance with the regulations for the Degree of Doctor of Philosophy

Information Security Institute
Faculty of Science and Technology
Queensland University of Technology

May 7, 2010
Keywords

Block cipher, stream cipher, symmetric cipher, linear transformation, diffusion, cryptanalysis, fixed points, round function, key scheduling algorithm, integral attack, bit-pattern, algebraic analysis, system of equations, branch number, AES, ARIA, LEX, BES, Noekeon, PRESENT, Serpent, SMS4
Abstract

This thesis is devoted to the study of linear relationships in symmetric block ciphers. A block cipher is designed so that the ciphertext is produced as a nonlinear function of the plaintext and secret master key. However, linear relationships within the cipher can still exist if the texts and components of the cipher are manipulated in a number of ways, as shown in this thesis.

There are four main contributions of this thesis. The first contribution is the extension of the applicability of integral attacks from word-based to bit-based block ciphers. Integral attacks exploit the linear relationship between texts at intermediate stages of encryption. This relationship can be used to recover subkey bits in a key recovery attack. In principle, integral attacks can be applied to bit-based block ciphers. However, specific tools to define the attack on these ciphers are not available. This problem is addressed in this thesis by introducing a refined set of notations to describe the attack. The bit pattern-based integral attack is successfully demonstrated on reduced-round variants of the block ciphers Noekeon, PRESENT and Serpent.

The second contribution is the discovery of a very small system of equations that describe the LEX-AES stream cipher. LEX-AES is based heavily on the 128-bit-key (16-byte) Advanced Encryption Standard (AES) block cipher. In one instance, the system contains 21 equations and 17 unknown bytes. This is very close to the upper limit for an exhaustive key search, which is 16 bytes. One only needs to acquire 36 bytes of keystream to generate the equations. Therefore, the security of this cipher depends on the difficulty of solving this small system of equations.

The third contribution is the proposal of an alternative method to measure diffusion in the linear transformation of Substitution-Permutation-Network (SPN) block ciphers. Currently, the branch number is widely used for this purpose. It is useful for estimating the possible success of differential and linear
attacks on a particular SPN cipher. However, the measure does not give information on the number of input bits that are left unchanged by the transformation when producing the output bits. The new measure introduced in this thesis is intended to complement the current branch number technique. The measure is based on fixed points and simple linear relationships between the input and output words of the linear transformation. The measure represents the average fraction of input words to a linear diffusion transformation that are not effectively changed by the transformation. This measure is applied to the block ciphers AES, ARIA, Serpent and PRESENT. It is shown that except for Serpent, the linear transformations used in the block ciphers examined do not behave as expected for a random linear transformation.

The fourth contribution is the identification of linear paths in the nonlinear round function of the SMS4 block cipher. The SMS4 block cipher is used as a standard in the Chinese Wireless LAN Wired Authentication and Privacy Infrastructure (WAPI) and hence, the round function should exhibit a high level of nonlinearity. However, the findings in this thesis on the existence of linear relationships show that this is not the case. It is shown that in some exceptional cases, the first four rounds of SMS4 are effectively linear. In these cases, the effective number of rounds for SMS4 is reduced by four, from 32 to 28. The findings raise questions about the security provided by SMS4, and might provide clues on the existence of a flaw in the design of the cipher.
## Contents

### Front Matter
- Keywords ................................................................. i
- Abstract ................................................................. iii
- Table of Contents ....................................................... v
- List of Figures ......................................................... xi
- List of Tables ......................................................... xiii
- List of Algorithms .................................................... xvii
- Declaration ........................................................... xix
- Previously Published Material ................................... xxi
- Acknowledgements .................................................. xxiii

### 1 Introduction
- 1.1 Linearity in Block Ciphers ...................................... 2
- 1.2 Aims and Contributions .......................................... 4
- 1.3 Outline of Thesis ................................................ 6

### 2 Symmetric Ciphers
- 2.1 Overview of Symmetric Ciphers .............................. 10
  - 2.1.1 Notation ..................................................... 11
  - 2.1.2 Block Cipher ............................................... 12
    Substitution-Permutation-Network .......................... 13
    Feistel Network .................................................. 13
    Modes of Operation ............................................ 14
  - 2.1.3 Stream Cipher .............................................. 15
- 2.2 Basics of Block Cipher Cryptanalysis ...................... 15
  - 2.2.1 Threat Model .............................................. 15
  - 2.2.2 Generic Attack Model .................................... 16
2.2.3 Attack Complexities

2.3 Existing Cryptanalysis Techniques

2.3.1 Linear Cryptanalysis

2.3.2 Differential Cryptanalysis

2.3.3 Truncated and Higher-Order Differentials

2.3.4 Impossible Differentials

2.3.5 Boomerang and Rectangle Attacks

2.3.6 Integral Attack

2.3.7 Slide Attack

2.3.8 Related-Key Attacks

2.3.9 Algebraic Cryptanalysis

2.4 Analyzed Block Ciphers

2.4.1 AES

2.4.2 ARIA

2.4.3 Noekeon

2.4.4 Serpent

2.4.5 PRESENT

2.4.6 SMS4

vi
## 3 Integral Attack on Bit-Based Block Ciphers

### 3.1 Bit-Pattern-Based Integral Attack

- **3.1.1 The Bit-Pattern-Based Notations**
- **3.1.2 Tracing the Bit Patterns**
  - Linear Transformation
  - Nonlinear Transformation
- **3.1.3 The Generic Attack**
  - Distinguishing Phase
  - Key Recovery Phase
  - Attack Extensions

### 3.2 Applications

- **3.2.1 Noekeon**
  - 3.5-round Distinguisher
  - 4-round Key Recovery Attack
  - 5-round Key Recovery Attack
- **3.2.2 Serpent**
  - 3.5-round Distinguisher
  - 4-round Key Recovery Attack
  - 5-round Key Recovery Attack
  - 6-round Key Recovery Attack
- **3.2.3 PRESENT**
  - 3.5-round Distinguisher
  - 4-round Key Recovery Attack
  - 5-round Key Recovery Attack
  - 6-round Key Recovery Attack
  - 7-round Key Recovery Attack

### 3.3 Experimental Results

- **3.3.1 Format of Experiments**
- **3.3.2 Discussion of Results**

### 3.4 Discussion
3.5 Related Work .............................................. 81
3.6 Summary and Conclusion ............................... 82

4 Algebraic Analysis of LEX-AES ......................... 85
  4.1 Preliminaries .............................................. 86
  4.2 Forming Equations to Describe LEX-AES .......... 87
    4.2.1 Keystream Generation Equations ................. 87
    4.2.2 Key Schedule Equations ........................... 93
    4.2.3 Additional Substitutions ......................... 95
    4.2.4 The Final System of Equations ................. 96
    4.2.5 Solving the Equations ............................. 96
    4.2.6 Alternative Methods for Obtaining Equations .... 98
  4.3 Forming Equations in Small Scale Variants of LEX-BES 99
    4.3.1 BES ................................................ 99
      Encryption Algorithm .................................. 101
      Key Scheduling Algorithm .............................. 102
    4.3.2 LEX-BES ........................................... 103
      Initialization .......................................... 103
      Keystream Generation ................................. 103
      Equation System for LEX-BES ....................... 104
    4.3.3 Small Scale LEX-BES .............................. 105
      Equation System for Small Scale LEX-BES .......... 105
    4.3.4 Experimental Results .............................. 106
  4.4 Discussion .............................................. 108
  4.5 Summary and Conclusion ............................. 109

5 Diffusion in the Linear Transformations of SPN Block Ciphers 111
  5.1 Preliminaries ........................................... 112
    5.1.1 Fixed Points in Random Permutations .......... 113
    5.1.2 Fixed Points in Linear Transformations ....... 113
      Rank of Random Matrices ............................. 114
      Rank of Matrices of the Type $A - I$ ............... 116
    5.1.3 Linear Diffusion Transformations using Nonsingular Matrices 117
  5.2 Measure of Diffusion Based on Fixed Points ........ 117
  5.3 Applications ......................................... 119
    5.3.1 AES ............................................... 120
## 6 Linearity within the SMS4 Block Cipher

6.1 Observations on Components in the Round Functions

6.1.1 Simple Linear Relationships between Input and Output
Words .................................................................................. 137
Nonlinear Transformation $S$ .................................................. 138
Linear Transformation $L$ ....................................................... 138
Function $T$ ............................................................................. 138
Linear Transformation $L'$ ...................................................... 140
Function $T'$ ........................................................................... 141
6.1.2 Relationship between $T$ and $T'$ ................................... 142
6.1.3 On the Branch Number of $L'$ ....................................... 142

6.2 Cryptographic Significance .................................................. 145
6.2.1 Implications for the Key Scheduling Algorithm ............... 145
6.2.2 Implications for the Encryption Algorithm ..................... 146
6.2.3 Further Implications for Both the Key Scheduling and the
Encryption Algorithms .......................................................... 147
6.2.4 Susceptibility to Algebraic Attack .................................. 148
6.2.5 Susceptibility to Advanced Variants of the Slide Attack ... 148
6.2.6 Subkeys and Related-Keys ............................................. 148

6.3 A Differential Attack on Modified SMS4 ............................. 150
6.3.1 23-Round Characteristic ............................................... 150
6.3.2 27-Round Key Recovery Attack .................................... 151
6.3.3 Comments on the Security of SMS4 ...................................... 153
6.4 Summary and Conclusion ..................................................... 153

7 Conclusions and Open Problems .............................................. 155

7.1 Review of Contributions ..................................................... 155

7.1.1 Chapter 3 ................................................................. 156
7.1.2 Chapter 4 ................................................................. 157
7.1.3 Chapter 5 ................................................................. 158
7.1.4 Chapter 6 ................................................................. 158

7.2 Open Problems ................................................................. 159

A S-Boxes ................................................................................. 163

B Difference Distribution Tables .............................................. 167

C LEX-AES Equations .............................................................. 173

C.1 Equations ........................................................................ 173

C.1.1 Forward Direction ......................................................... 173
C.1.2 Backward Direction ....................................................... 175
C.1.3 Subkey Variables Substitution ........................................ 177
C.1.4 Temporary Variables ..................................................... 182
C.1.5 Final Equation System ................................................... 184

C.2 Step-by-Step Procedure to Produce Equations for Substitution 204

Bibliography ............................................................................ 207
# List of Figures

2.1 A generic symmetric cipher ........................................... 10
2.2 Generic structure of an $R$-round block cipher $E$ ............... 12
2.3 A boomerang distinguisher ........................................... 24
2.4 A 3-round integral distinguisher for the AES .................... 30
2.5 Round function of the AES in Round $r$ .......................... 34
2.6 Round function and nonlinear transformations of ARIA in Round $r$ 39
2.7 Round function of Noekeon in Round $r$ .......................... 43
2.8 Round function of Serpent in Round $r$ .......................... 45
2.9 Round function of PRESENT in Round $r$ .......................... 48
2.10 Round function of SMS4 in Round $i$ ............................ 51
2.11 Initialization and keystream generation of LEX-AES ........... 53
2.12 Different leaks in odd and even rounds ......................... 55

3.1 Example of the round function of a bit-based block cipher ...... 58
3.2 Example of the round function of a word-based block cipher ... 58
3.3 The 3.5-round bit-pattern-based integral distinguisher for Noekeon. 67
3.4 The 3.5-round bit-pattern-based integral distinguisher for Serpent. 70
3.5 The 3.5-round bit-pattern-based integral distinguisher for PRESENT. 74

4.1 State byte variables and constants involved in building the system of equations. ............................................... 87
4.2 Subkey byte variables. ............................................... 88
4.3 Example of forming equations to describe two constants: one before and one after the middle iteration. Known keystream bytes (constants) are denoted in gray. ................................. 89
4.4 Keystream involved in blocks of 10 rounds. ...................... 97
List of Tables

2.1 Example of text values for a structure of sixteen text blocks which has the property ACCB .............................................. 26
2.2 Summary of existing attacks on the AES ................................... 38
2.3 Summary of existing attacks on ARIA .................................... 41
2.4 The linear transformation of Serpent. The output bits $\tilde{X}_{r+1}$ are displayed as the XOR sum of the input bits $\tilde{Z}_j$. For instance, $\tilde{X}_{96} = \tilde{Z}_9 \oplus \tilde{Z}_{86} \oplus \tilde{Z}_{121}$. .............................................. 46
2.5 Summary of existing attacks on Serpent ................................ 47
2.6 Summary of existing attacks on Present ................................ 49
2.7 Summary of existing attacks on the SMS4 block cipher ................. 52
3.1 Examples of text values for $c$, $a_i$ and selected $b_i$ bit patterns in a structure of sixteen texts where $x, y \in \mathbb{F}_2$ and $y = \bar{x}$ ..................... 60
3.2 Example of 8-bit text values denoted by the pattern $a_0a_3c_2c_1c_1c_0$ .... 61
3.3 Example of how a 4-bit text structure denoted by the pattern $c_3c_2$ evolves into $b_2b_2b_2b_2$ through a bijective $4 \times 4$ S-box $s$ ......... 63
3.4 Key bit positions $j$ involved in the 5-round attack ...................... 72
3.5 Bit patterns of $Z_0$ in the 6-round attack ............................... 73
3.6 Summary of attacks .................................................. 77
3.7 Experimental results of bit-pattern based integral attack to Noekeon, Serpent and PRESENT reduced to 4 rounds. ......................... 79
4.1 Comparison between BES$(n, r, c, e)$ and LB$(n, r, c, e)$ in terms of the number of equations and variables. ......................... 105
4.2 Number of equations and variables for LB$(10, 2, 2, 4)$ ............... 106
4.3 Number of equations and variables for small scale BES defined over $GF(2^4)$ .................................................. 106
4.4 Time (in seconds) and memory (in MB) required to compute Gröbner basis for the equation system arising from LB(1, 2, 2, 4).

4.5 Time (in seconds) and memory (in MB) required to compute Gröbner basis for the equation system arising from LB(10, 2, 2, 4).

5.1 Theoretical probabilities that random permutations of \( n \) elements have \( c \) fixed points.

5.2 Theoretical probabilities \( \hat{p}_{2,m,r} \), that \( m \times m \) matrices over \( \mathbb{F}_2 \) have rank \( r \).

5.3 Theoretical probabilities \( \hat{p}_{2^8,m,r} \), that \( m \times m \) matrices over \( \mathbb{F}_{2^8} \) have rank \( r \).

5.4 Experimental estimates of probabilities of \( r = \text{rank}(A - I) \), where \( A \) is a nonsingular \( m \times m \) matrix over \( \mathbb{F}_q \).

5.5 Parameters of \( L \).

5.6 Number of input blocks to the transformation \( L \) of the AES such that \( AZ = I(0)Z \).

5.7 Number of input blocks to the transformation \( L \) of ARIA such that \( AZ = I(0)Z \).

5.8 Number of input blocks to the transformation \( L \) of PRESENT such that \( AZ = I(0)Z \).

5.9 Number of input blocks to the transformation \( L \) of Serpent such that \( AZ = I(0)Z \).

5.10 Summary of the observations on the transformation \( L \).

5.11 Permutation of bits in the modified \( L \) transformation of PRESENT. \( P(i) \) means bit \( i \) is moved to position \( P(i) \) after the transformation.

6.1 Values of \( X_i \) (in the set \( \Theta_S \)) and \( j \) such that \( S(X_i) = X_i \ll j \).

6.2 Number of output words which are equivalent to the rotation of the input word by \( j \) bits to the left (\( 0 \leq j \leq 31 \)), for each component function.

6.3 Values of \( X_i \) (in the set \( \Theta_T \)) and \( j \) such that \( T(X_i) = X_i \ll j \).

6.4 Values of \( X_i \) (in the set \( \Theta_{T'} \)) and \( j \) such that \( T'(X_i) = X_i \ll j \).

6.5 The input-output pattern distribution of \( L' \).

A.1 The S-box table of the AES and \( s_1 \) of ARIA.

A.2 The S-box \( s_2 \) table of ARIA.

A.3 The S-box of Noekeon.
A.4 The S-box of Present ........................................ 165
A.5 The eight S-boxes of Serpent ............................... 165
A.6 The S-box table of SMS4 .................................... 165

B.1 The Difference Distribution Table of the S-box of Noekeon ... 168
B.2 The Difference Distribution Table of the S-box s₀ of Serpent ... 168
B.3 The Difference Distribution Table of the S-box s₁ of Serpent ... 169
B.4 The Difference Distribution Table of the S-box s₂ of Serpent ... 169
B.5 The Difference Distribution Table of the S-box s₃ of Serpent ... 170
B.6 The Difference Distribution Table of the S-box s₄ of Serpent ... 170
B.7 The Difference Distribution Table of the S-box s₅ of Serpent ... 171
B.8 The Difference Distribution Table of the S-box s₆ of Serpent ... 171
B.9 The Difference Distribution Table of the S-box s₇ of Serpent ... 172
B.10 The Difference Distribution Table of the S-box of Present ... 172
# List of Algorithms

2.1 Algorithm for recovering \( k \) bits of the last round subkey in a generic \( R \)-round key recovery phase .................................................. 18
2.2 Algorithm to calculate the Linear Approximation Table (LAT) ... 20
2.3 Algorithm to calculate the Difference Distribution Table (DDT) ... 22
2.4 Algorithm for constructing an integral distinguisher ..................... 28
2.5 Algorithm for \( R \)-round key recovery phase in an integral attack ... 29
2.6 Key Scheduling Algorithm for the AES. ...................................... 36
3.1 Algorithm for basic attack .......................................................... 65
3.2 Algorithm for conducting the bit-pattern-based integral attack ex-periments ................................................................. 78
4.1 Computing the elements of the matrix \( L_M \) ................................. 103
Declaration

The work contained in this thesis has not been previously submitted for a degree or diploma at any higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed: ........................................... Date: ............................
Previously Published Material

The following papers have been published or presented, and contain material based on the content of this thesis.


Acknowledgements

Firstly, I would like to thank God for giving me the opportunity to do my PhD at one of the most prestigious security institute, the Information Security Institute (ISI), Queensland University of Technology (QUT). I thank my Principal Supervisor, Prof. Ed Dawson, whose encouragement, support and guidance allowed me to complete my PhD journey. I am grateful to my Associate Supervisors Dr. Leonie Simpson, Dr. Matt Henricksen, Dr. Håvard Raddum and Dr. Kenneth Wong, for giving constructive comments and support to my work. I thank the internal review panel members and the external examiners for providing valuable comments that improve the presentation of this thesis. Apart from my supervisors, the internal review panel comprises of Dr. Gary Carter and Dr. Juan Manuel González Nieto. My thanks also goes to Dr. Subariah Ibrahim from Universiti Teknologi Malaysia and Dr. Raphael Phan from Loughborough University, UK (previously in Swinburne University of Technology, Sarawak campus, Malaysia) for introducing me to the world of cryptography. I acknowledge my family for their moral support and encouragement throughout my PhD. I also give my sincere thanks to all members of the ISI and friends.

I am grateful to QUT’s High Performance Computing (HPC) & Research Support group for giving me access to their supercomputer that allowed me to conduct various experiments. I thank QUT and the ISI for giving me the much needed funds to support my trip to various prestigious conferences all around the world. I also would like to extend my gratitude to the Malaysian Institute of Microelectronic Systems (MIMOS) and the Malaysian Ministry of Science, Technology and Innovation (MOSTI) for providing financial support during the course of my PhD.
Chapter 1

Introduction

In the world of information technology today, symmetric block ciphers are undeniably paramount in providing confidentiality, data integrity, authentication and verification [133]. For instance, the Advanced Encryption Standard (AES) block cipher was initially approved by the United States (US) government to be used in protecting sensitive but unclassified digital information. However, in June 2003, the Committee on National Security Systems (CNSS), which is a committee under the administration of US Department of Defense, broadened the AES scope to include protecting classified information up to SECRET level [50]. For TOP SECRET level, a key size of either 192 or 256 bits is required. The IDEA [121] block cipher has been included in many cryptographic applications such as Pretty Good Privacy (PGP) and Secure Shell (SSH). The block cipher Kasumi is used for securing mobile communications within the 3rd Generation Partnership Project (3GPP) [1]. In March 2007, Sony Corporation announced the use of the block cipher CLEFIA for use in advanced copyright protection and authentication [164].

The plethora of block ciphers in the market today demands significant attention from the cryptographic community. Extensive analysis of a cipher which include simplified variants, reduced-round versions and alternate implementations are crucial to ensure that its security remain current. If a cipher is shown conclusively to be vulnerable against an attack with reasonable complexity, then this will cast serious doubts about its security. Furthermore, an attack which was infeasible in the past may become plausible now or in the near future due
to rapid developments of technology. This can be attributed to the so-called Moore’s Law which states that the processing power of computers doubles every 18 months \[134,135\]. The need to ensure that the security of block ciphers remain current is the main motivation of conducting the investigation in this thesis.

This thesis investigates linear relationships in block ciphers. The investigation includes the analysis of block cipher components used in stream ciphers. The output of a block cipher is a nonlinear function of the input and key. Despite this, it will be shown in this thesis that linearity within a cipher can still be detected if the input texts and components of the cipher are manipulated in certain ways. Furthermore, for particular inputs, the nonlinear components will also be shown to exhibit some linear behaviour.

1.1 Linearity in Block Ciphers

Linearity exists in block ciphers mainly due to the linear transformation. One of the earliest known ciphers that uses only linear transformations is the Caesar cipher \[93\]. In its original form, the ciphertext letter is produced by shifting the original plaintext letter by three to the right. Therefore, applied to the English alphabet, the plaintext letter ‘A’ is encrypted to form the ciphertext ‘D’, ‘B’ to ‘E’, and so forth. However, ciphers using only linear transformations are vulnerable to algebraic attacks. For example, the Caesar cipher can be easily broken if only the ciphertext message is available to the attacker. The attacker can just shift all ciphertext letters by three letters to the left, which is a linear operation, to obtain the original plaintext message.

In modern block ciphers, the linear transformation is a vital component that provides the diffusion property introduced by Shannon \[162\]. According to Shannon, each bit of the output block should depend on each bit of the input block. Most linear transformations use at least one of these operations: modular addition and subtraction, XOR, rotation and shift.

The other important property that a block cipher should have is confusion. Confusion is needed in a block cipher so that the ciphertext is related to the plaintext and master key in a highly nonlinear way. A nonlinear transformation can be implemented in a block cipher using S-boxes. An S-box basically substitutes one word for another word in a nonlinear manner. The combination of diffusion and confusion are essential to thwart statistical attacks.
1.1. Linearity in Block Ciphers

Several methods for measuring diffusion have been proposed. One of these methods is the *avalanche effect* proposed by Feistel [75]. A component has the avalanche effect if complementing one input bit causes half of the output bits to change, on average. If every output bit of a component depends on all input bits, then the component has the *completeness* property, which was introduced by Kam and Davida [94]. These two concepts were combined by Webster and Tavares to define the *strict avalanche criterion (SAC)* [175]. A component adheres to the SAC if complementing one input bit causes every output bit to change with probability 0.5.

For security, meeting the avalanche criteria may be necessary but it is not sufficient. As pointed out by Rijmen [155], the drawbacks of these properties are that they do not address the case of a large change at the input, are probabilistic and concerned only with resistance to differential cryptanalysis. The *branch number*, introduced by Daemen, is an attempt to address these problems. It is a measure of the minimum number of active S-boxes for a Substitution-Permutation Network (SPN) block cipher for any two consecutive rounds [58, 62].

Most of the time, complete diffusion is not achieved in a single round. If a cipher has weak one-round diffusion, then this shortcoming is normally compensated for by employing a high number of rounds. Example of ciphers which have this property are 48-round CAST-256 [2], 32-round SMS4 [64, 149] and 31-round PRESENT [37]. It is believed that more rounds mean more security. However, a large number of rounds has performance implications. Therefore, cipher designers need to achieve a balance between performance and security.

One also needs to consider the platform in which the cipher is to be implemented. For instance, a single $8 \times 8$ S-box lookup on a 32-bit machine requires four assembly code operations in the Intel Pentium 4 architecture [84]. If a larger-sized S-box is used, then more operations are needed and thus, this penalizes the performance of the cipher. This is why linear transformations which operate using inexpensive operations such as XOR, rotations and shifts, are needed to balance the necessity of nonlinearity by using one or more S-boxes.

The use of a combination of linear and nonlinear transformations make the relationship between inputs and outputs of the cipher nonlinear. Despite this, in certain cases, linearity can still be present. The linearity is revealed by manipulating any of these parameters: the plaintext, master key and ciphertext blocks. The linearity resulting from this manipulation can be used to launch attacks on
block ciphers. Almost all attacks are based on the existence of some linear relationships. For instance, in 1985, Shamir discovered that there are many entries of the S-box of the Data Encryption Standard (DES) that have output parity of zero \[161\]. At the time, the significance of the findings were unknown since the design criteria of the DES were not made public. About eight years later, in 1993, this property was exploited by Matsui in an attack which is now known as linear cryptanalysis \[132\]. The attack is not only applicable to the DES but to a wide range of ciphers. It is now standard practice for every block cipher to demonstrate its resistance to this type of attack.

Other attacks that manipulate linear relationships are all attacks based on differential cryptanalysis \[23,24\]. In its basic form, the attack exploits the existence of the XOR difference between two ciphertexts that occurs with nontrivial probability. This includes truncated differentials \[108\], impossible differentials \[13\], boomerang \[173\], rectangle \[14,16\], integral \[59,112\] and slide \[35\] attacks.

There are also attacks that manipulate the existence of a subset of keys that cause the full cipher to behave linearly. For instance, for a particular set of weak master keys, some plaintext blocks are fixed points for the 16-round DES \[51\], 16-round Blowfish \[97\], 32-round GOST and 6- and 8-round DEAL \[96\] and 528-round of KeeLoq \[54\] block ciphers. In the specific case of KeeLoq, the cipher uses an excessive number of rounds to alleviate the weakness inherent in its round function. In these cases, for a particular set of inputs, the cipher is distinguishable from a random permutation and the distinguisher can be used to recover subkey bits in attacks.

### 1.2 Aims and Contributions

The main aim of this thesis is to investigate linear relationships in block ciphers. The existence of nonlinear components in the round function of block ciphers does not necessarily eliminate the linear relationships in these ciphers. Linearity can still exist under certain circumstances, and is revealed if the input texts and components of the cipher are manipulated in certain ways. The investigation in this thesis also incorporates the analysis of block cipher components used in stream ciphers. This thesis has four major contributions:

\[1\]DEAL uses six rounds if the size of the master key is 192 bits and eight rounds if the master key is 256 bits.
1. **Extension of the applicability of integral attacks from word-based to bit-based block ciphers.** The integral attack, which is an attack that manipulates the existence of a linear relationship between the texts at intermediate stages of encryption, is extended. Traditional integral attacks are best suited for word-based block ciphers. In principal, the attack can be applied to bit-based block ciphers, however, no generic method is available in the literature. This gap is addressed in this thesis with the proposal of the bit pattern-based integral attack in Chapter 3. The improved attack is a generic method of applying the integral attack to bit-based block ciphers. The attack is shown to be successfully applied to reduced-round versions of the block ciphers Noekeon, Serpent and PRESENT. This is the first time that a generic method is proposed in applying integral attack to bit-based block ciphers.

2. **Identification of a very small system of equations that describe the LEX-AES stream cipher.** This contribution involves the investigation of algebraic interaction between block cipher components used in the LEX-AES stream cipher. The investigation reveals that a very small system of equations can be constructed that describes LEX-AES. The equations contain linear and nonlinear relationships that involve the description of the keystream as a function of the internal state of the cipher. The number of variables in the system is very close to the threshold for a key recovery attack. Due to this, it is believed that an attack that manipulates these equations is of similar strength to an attack that is one or two rounds short of breaking the full cipher. The amount of keystream required to generate the equations is very small and reasonably practical to obtain in the real world. The results of this investigation, which are presented in Chapter 4, are the first to explore these relationships using the LEX method.

3. **Proposal of a new method of measuring diffusion in SPN block ciphers.** One of the most widely used measures of the diffusion provided by the linear transformation of SPN block ciphers is the branch number. It can be used as a tool to approximate the cipher’s strength against linear and differential attacks. However, it does not provide an indication of how well the linear transformation effectively changes the value of the input block when producing the output block. This problem is addressed with the proposal of a method that incorporates simple linear relationships between the input
and output bits of the linear transformation. The method also provides indication whether a linear transformation has other, much simpler representation for particular input blocks. If many such representations exist, then the cipher might be vulnerable to attacks. The method is applied to the block ciphers AES, ARIA, Serpent, Present. In particular, it is shown that except for Serpent, the linear transformations of the block ciphers examined do not behave as expected for a random linear transformation. A full discussion of this new proposed method is given in Chapter 5.

4. Identification of linear paths in nonlinear components of the SMS4 block cipher. This contribution involves the study of simple linear relationships between the input and output of the components of the SMS4 block cipher. The resulting relationships reveal new and unexpected properties of the components that have the potential to be exploited in attacks. In particular, it is discovered that one of the nonlinear functions used in SMS4 does not behave as expected for a random permutation. It is also shown that in some exceptional cases, the security of the 32-round SMS4 block cipher can be theoretically reduced to 28 rounds. The results of this investigation are presented in Chapter 6.

1.3 Outline of Thesis

This thesis is organized as follows.

- Chapter 2 describes the background information necessary to provide the context for the subsequent chapters. The information includes the basics of symmetric ciphers, cryptanalytic techniques and the description of the specific block ciphers analyzed in this thesis.

- Chapter 3 introduces the new bit-pattern based integral attack. This chapter provides the necessary tools to apply the existing integral attack on bit-based block ciphers. Some of the results in this chapter are published in the proceedings of the Fast Software Encryption (FSE) workshop 2008 [180].

- Chapter 4 presents an algebraic analysis of a block-cipher-based stream cipher called LEX-AES. In particular, it is shown that this cipher can be described using a very small overdefined system of equations with very few
unknowns. Some of the results in this chapter are published in the proceedings of the Australasian Information Security Conference (AISC) 2009 [181] and the 2nd International Cryptology Conference 2010 (Cryptology 2010), Malaysia [182].

- Chapter 5 introduces a proposed new measure of diffusion in the linear transformation of SPN block ciphers. In this chapter, this measure is applied to the SPN block ciphers AES, ARIA and PRESENT and Serpent; and the security implications are investigated.

- Chapter 6 presents the results of the analysis of the linearity within the SMS4 block cipher. It is determined that for every component used in this cipher, there exists a very simple linear relationship between particular values of input and output. Some of the results in this chapter will be published in the proceedings of the International Conference on Information Security and Cryptology (Inscrypt) 2009 [179].

- Chapter 7 presents conclusions about the work presented in this thesis. In addition, some future directions for exploration are proposed.

- Appendix A contains the details of the S-boxes for all block ciphers examined in this thesis.

- Appendix B provides the Difference Distribution Tables (DDTs) corresponding to the S-boxes of the block ciphers mentioned in Chapter 3.

- Appendix C contains the list of equations arising from the LEX-AES stream cipher investigated in Chapter 4.
Chapter 2

Symmetric Ciphers

There has been extensive research into symmetric ciphers, particularly block ciphers, in the last 40 years. Before the 1970s, symmetric ciphers were used mostly in military and government circles. Their use can be traced back to the times of Julius Caesar, and more recently, during World Wars I and II [9, 58, 93].

During the 1970s, symmetric ciphers began to be used increasingly for commercial applications. This is attributed to the introduction of the Data Encryption Standard (DES) in 1977 by the then National Bureau of Standards (NBS) [141] (now National Institute of Standards and Technology (NIST)). Although NIST is a United States body, the cipher became a de facto global standard. The financial sector was one of the first to use the DES [58]. By the late 1990s, the 56-bit key size of the DES was vulnerable to exhaustive key search using custom-built hardware and a large network of computers [71, 73, 117].

In 1997, the NIST called for an open evaluation process [122, 122] for a cipher to replace the DES. In 1998, fifteen block ciphers were received as candidates. After two years of public scrutiny, Rijndael [62] was selected as the new Advanced Encryption Standard (AES) [144] in October 2000. The AES sparked similar efforts in other countries such as the Cryptography Research and Evaluation Committee (CRYPTREC) [57] in Japan. In Europe, processes included the New European Schemes for Signatures, Integrity and Encryption (NESSIE) [147]; and eSTREAM, the ECrypt Stream Cipher Project [6]. In addition to these schemes, ciphers are also proposed in conferences and journals for public scrutiny (e.g. [92, 116, 118, 123, 138, 164]). The openness of these forums gives a clear indi-
cation of the importance of public analysis in evaluating the security of ciphers.

This chapter presents the background information necessary to provide the foundation for subsequent chapters. It is organized as follows. Section 2.1 presents an overview of symmetric ciphers. Concepts related to cryptanalysis are given in Section 2.2. Section 2.3 describes the most relevant cryptanalysis techniques and Section 2.4 introduces the block ciphers which will be analyzed in the subsequent chapters of this thesis.

## 2.1 Overview of Symmetric Ciphers

A symmetric cipher is an algorithm that is capable of transforming a message into an unreadable form, and vice-versa, using the same secret master key. The original message is called the *plaintext*, denoted $P$. The unreadable form is called the *ciphertext*, denoted by $C$. The secret master key is denoted by $K$. The transformation of $P$ into $C$ is called *encryption* and the reverse process is called *decryption*. If $E_K$ denotes the encryption algorithm using the master key $K$, then the encryption and decryption processes can be described as follows:

$$E_K(P) = C, \quad E_K^{-1}(C) = P$$

Figure 2.1 illustrates the encryption and decryption processes required when a plaintext $P$, requiring confidentiality, is passed from sender to receiver. Encryption and decryption are performed using a symmetric cipher. The figure shows that the ciphertext is sent to the receiver via an insecure communication channel (indicated by the dashed line). Before communication can begin, both the sender and receiver need to agree on a secret master key. The key is then distributed securely between both parties. Methods of distributing the secret master keys securely are not discussed in this thesis.

![Figure 2.1: A generic symmetric cipher](image-url)
2.1. Overview of Symmetric Ciphers

2.1.1 Notation

In addition to the notation introduced above (where \( P \), \( C \) and \( K \) denote the plaintext, ciphertext and master key blocks, respectively), the following notations will be used consistently throughout this thesis:

- \( K^r \) is the subkey used in round \( r \).
- \( X^r \) is the input block to round \( r \) where \( X^0 = P \) is the plaintext block.
- \( Y^r \) is the output block of the key mixing transformation in round \( r \).
- \( Z^r \) is the output block of the nonlinear transformation in round \( r \).
- The \( mb \)-bit block \( W^r = (W^r_0, W^r_1, \ldots, W^r_{m-1}) \) is formed from the concatenation of \( m b \)-bit words \( W^r_i \) where the value \( b \) should be clear from the text.
- \( S_i \) denotes the nonlinear transformation of the \( i \)-th round function. The subscript is omitted if there is only a single nonlinear transformation.
- \( L_i \) denotes the linear transformation of the \( i \)-th round function. The subscript is omitted if there is only a single linear transformation.
- The operator \( \oplus \) denotes XOR; \( W \ll i \) and \( W \gg i \) denote the rotation of the word \( W \) by \( i \) bits to the left and right, respectively; \( W \ll i \) and \( W \gg i \) denote the shift of the word \( W \) by \( i \) bits to the left and right, respectively.
- Hexadecimal numbers are written in teletype font. For instance, the number \( B \) is the hexadecimal representation for the decimal number 11.
- In describing data complexities for attacks, \( KP \) denotes known plaintexts, \( CP \) denotes chosen plaintexts, \( ACP \) denotes adaptive chosen plaintexts, \( ACC \) denotes adaptive chosen ciphertexts and \( RK \) denotes related-keys.

Notation for which usage is confined to one chapter or section only will be introduced in that chapter or section.
2.1.2 Block Cipher

In general, a block cipher accepts a \( mb \)-bit plaintext block \( P \) and a \( \hat{mb} \)-bit secret master key \( K \). The master key is used as input to the key scheduling algorithm to produce a set of \( R \) round subkeys. The plaintext block and the round subkeys are used as input to the encryption algorithm to produce the ciphertext block \( C \). The encryption and decryption algorithms consist of \( R \) applications of the round function where \( R \) is referred to as the number of rounds of the cipher.

Let \( P = (X_0^0, X_1^0, \ldots, X_{m-1}^0) \) denote the \( mb \)-bit plaintext block formed from the concatenation of \( m \) \( b \)-bit words \( X_i^0 \). Let \( K^r \) denote the subkey in round \( r \) derived from the \( \hat{mb} \)-bit master key \( K \). Let \( F_{K^r} \) denote the round function in round \( r \) composed of the nonlinear transformation \( S \) and the linear transformation \( L \). These transformations should provide the confusion and diffusion properties required in a cipher \[162\]. The generic structure of an \( r \)-round block cipher is depicted in Figure 2.2.

The encryption round function \( F_{K^r} \) is composed of nonlinear and linear transformations. The linear transformation is comprised of the key mixing and diffusion transformations. The key mixing transformation adds the round subkey to the current state block using linear operations, such as XOR and addition modulo \( 2^{32} \). The diffusion transformation operates such that each word of the output block linearly depends on many words of the input block. The nonlin-
The round function \( F_{K_{r+1}}^{−1} \) for decryption contains the inverse of the transformations in encryption.

The key scheduling algorithm accepts the secret master key block \( K \) to produce the round subkey \( K_r \) for each round. Similar to the plaintext block, the master key block is subjected to some linear and nonlinear transformations before the round subkey is produced for every round. The value of a round subkey typically depends on a subset of previous round subkeys. The algorithm for the key scheduling is normally different from the encryption and decryption algorithms.

Two main types of block ciphers used today are the Substitution-Permutation-Network (SPN) and the Feistel network. Another type of block cipher is the Lai-Massey scheme \([118, 120, 138, 171]\), however, this type is not investigated in this thesis. The SPN and Feistel networks are described briefly as follows.

**Substitution-Permutation-Network**

The round function of an SPN block cipher operates on the entire data block in each round. A typical round function in round \( r \) is denoted as follows:

\[
X^{r+1} = F_{K_{r+1}}^{−1}(X^r) = L(S(X^r \oplus K^r)), \quad r = 0, 1, \ldots, R - 1
\]

where the nonlinear transformation \( S \) is interleaved with the key addition and the linear diffusion transformation \( L \). This structure is used in many block ciphers such as the AES \([62, 144]\), ARIA \([116, 146]\), Present \([37]\), Serpent \([4, 12]\), Anubis \([8]\), Hierocrypt \([151]\) and SQUARE \([59]\) and has received much public scrutiny (e.g. \([7, 43, 45, 87, 95, 98]\)). In particular, the SPN block ciphers AES, ARIA, Present and Serpent are analyzed in Chapters \([3, 4]\) and \([5]\).

**Feistel Network**

In contrast with the SPN, the round function of a Feistel cipher operates on a subset of the data block in each round. In a traditional Feistel network, the plaintext block \( P = (X_0^0, X_1^0) \) is composed of two \( b \)-bit words \( X_i^r \). The round function \( F_{K_r} \) is applied only to one word in every round. The updated value of both words are swapped before they are used as input to the subsequent round.
as follows:

$$X^r = (X^{r-1}_1, X^{r-1}_0 \oplus F_{K^r}(X^{r-1}_1)), \quad r = 0, 1, \ldots, R - 1.$$ 

The above structure can be extended to the case where the plaintext block $P = (X^0_0, X^0_1, \ldots, X^0_{m-1})$ is composed of $m$ b-bit words $X^r_i$ as follows

$$X^r = (X^{r-1}_1, \ldots, X^{r-1}_0 \oplus F_{K^r}(X^{r-1}_1, \ldots, X^{r-1}_{m-1})), \quad r = 0, 1, \ldots, R - 1$$

where $m > 2$. This structure was used in the 1970s by the Data Encryption Standard (DES) [141] and by Lucifer [166] (the predecessor to the DES). Since then, the structure has been extensively analyzed (e.g. [66, 86, 107, 140, 159, 163, 172]). It is still in use in block ciphers today, including Camellia [5], CLEFIA [164], MARS [41] and Twofish [160].

Schneier and Kelsey provide a taxonomy of the various Feistel networks where the input words to the round function are referred to as the *source* block and the remaining words are called the *target* block [159]. The SMS4 [64, 149] block cipher, which according to the terminology of Schneier and Kelsey, is a homogeneous, complete, source-heavy unbalanced Feistel network [159]. Homogeneous means that the round function is identical in every round (except the round subkey). Complete means that in every round, each bit of the block is either in the target block or the source block. Source-heavy means that the size of the source block is larger than the target block. An unbalanced Feistel network means that the size of the source and target blocks are different. In this thesis, the SMS4 block cipher is investigated and the results are presented in Chapter 6.

**Modes of Operation**

Commonly, the plaintext to be encrypted is larger than the input block size for the block cipher. The plaintext is divided into appropriately sized blocks, which are then used as input to the encryption function. Similarly, the ciphertext is divided into blocks which are used as input to the decryption function. Several generic modes of operation can be used for encryption and decryption. These include the electronic codebook (ECB), cipher block chaining (CBC), cipher feedback (CFB), output feedback (OFB) and counter (CTR) modes.

This thesis considers only the case where the block ciphers operate in ECB mode. In this case, the ECB mode encrypts the plaintext one block at a time.
using the same secret master key independently of other plaintext blocks. The interested reader is referred to Schneier [158], Menezes, Oorschot and Vanstone [133], and Stinson [167] for the explanation of other modes of operation.

### 2.2 Basics of Block Cipher Cryptanalysis

This section introduces elementary concepts of cryptanalysis which provide the context for subsequent chapters.

#### 2.2.1 Threat Model

The threat model for a block cipher is based on Kerckhoffs’ assumption whereby the attacker knows the details of the cipher but not the secret master key [102]. The secrecy of the messages therefore lies entirely with the key. Based on this assumption, attacks can be classified according to an attacker’s capabilities:
• **Ciphertext-only attack.** An attacker possesses a set of ciphertext blocks encrypted by a cipher using the same master key.

• **Known-plaintext attack.** An attacker knows some plaintext-ciphertext pairs encrypted with the same key.

• **Chosen-plaintext attack.** An attacker is able to choose the plaintexts to be encrypted and obtain the corresponding ciphertexts.

• **Chosen-ciphertext attack.** An attacker is able to choose the ciphertexts to be decrypted and obtain the corresponding plaintexts.

• **Adaptive chosen-plaintext attack.** An attacker is able to choose plaintexts to be encrypted and modify subsequent plaintext choices based on previously obtained ciphertexts.

• **Adaptive chosen-ciphertext attack.** An attacker is able to choose ciphertexts to be decrypted and modify subsequent ciphertext choices based on previously obtained plaintexts.

• **Related-key attack.** An attacker is able to choose relationships between different keys used in encryption and decryption, but not the actual value of the keys.

Related-key attacks involve multiple secret master keys while the rest assume the use of a single secret master key. Adaptive chosen-plaintext/ciphertext attacks are the most powerful attacks but the ciphertext-only attack is the most practical form of attack. A cipher which can be shown to be secure against an adaptive chosen plaintext/ciphertext attack is also secure against chosen-plaintext/ciphertext, known-plaintext and ciphertext-only attacks [109]. A cipher which is vulnerable to a ciphertext-only attack is considered very weak.

### 2.2.2 Generic Attack Model

All attacks described later in Section 2.3, except for algebraic, are performed in two main phases: distinguishing and key recovery. Phases for algebraic attacks will be explained in Section 2.3.9.
Distinguishing Phase

In the distinguishing phase, the attacker constructs an algorithm that can be used to distinguish the output of the analyzed block cipher from the output of a random permutation. This algorithm is called a distinguisher. Normally, a distinguisher is constructed for a reduced-round variant of the block cipher. Suppose that the attacker wants to attack an $R$-round cipher. So the attacker needs to find an $\hat{R}$-round distinguisher where $\hat{R} < R$.

Key Recovery Phase

In a basic $R$-round key recovery attack using an $\hat{R}$-round distinguisher where $\hat{R} = R - 1$, an additional round is appended at the end of the $\hat{R}$-round distinguisher. Then, pairs of plaintext and ciphertexts after $R$ rounds are obtained. If $k$ bits of the $R$-th round subkey need to be recovered, then the affected subkey word is guessed. Next, for every guessed value, one-round partial decryptions of the affected ciphertext words are performed to obtain the output of round $\hat{R}$. If the resulting partial decryption (output of round $\hat{R}$) matches the conditions predicted by the distinguisher, then the guessed value is recorded as one of the possible correct subkey values. The most probable subkey is determined by repeating the attack with different plaintexts or ciphertexts. The key recovery phase is outlined in Algorithm 2.1.

2.2.3 Attack Complexities

The complexities of an attack on block ciphers are measured using the following parameters:

- **Data complexity.** The amount of plaintext or ciphertext blocks required to execute the attack.

- **Time complexity.** The number of operations needed to perform the attack. The operation may refer to encryption, decryption, or memory access.

- **Memory complexity.** The amount of memory needed to store the information required to perform the attack.

\(^1\)Note that if a distinguisher can be constructed for the full-round cipher, then the cipher has a serious security flaw.
begin
Input: $\hat{R}$-round distinguisher;
while correct subkey has yet to be determined do
    Generate or collect pairs of plaintext and ciphertext blocks that
    matches distinguisher;
    for subkey guess $v = 0$ to $2^k - 1$ do
        Partially decrypt ciphertexts using value $v$ as partial subkey
        bits to find output bits of round $r$;
        if conditions predicted by the distinguisher holds then
            Record $v$ as a possible correct subkey;
        end
    end
end
Output: value $v$ as correct subkey bits;
end

Algorithm 2.1: Algorithm for recovering $k$ bits of the last round subkey in a
generic $R$-round key recovery phase

- Number of rounds. The number of block cipher rounds penetrated by
the attack.

The main aim of a cryptanalyst is to develop an attack which can penetrate
as many rounds of a cipher as possible using the least possible data, time and
memory requirements. Typically, there are trade-offs associated with improving
each type of complexity. For instance, lowering the time complexity might involve
increasing the memory complexity at the same time.

In comparing various attacks on a particular block cipher, the following rules
are generally used. If two different attacks penetrate the same number of rounds,
then the attack which require the least amount of data, time and memory is better
than the other attack. If, however, two different attacks penetrate different
number of rounds, then the attack that manages to penetrate more rounds is
better than the other attack.

2.3 Existing Cryptanalysis Techniques

This section describes existing block cipher cryptanalysis techniques which are
useful in understanding the subsequent chapters. The attacks exploit linear
relationships between different plaintexts, ciphertexts and master key blocks.
2.3.1 Linear Cryptanalysis

Linear cryptanalysis is a known plaintext attack introduced by Matsui [132] in 1993. However, the origins of the attack can be traced back to the observation regarding the S-box of the DES made by Shamir [161] in 1986. In a basic linear attack, a one-bit linear relationship between selected bits of the plaintext block $P$, ciphertext block $C$ and master key block $K$ is constructed as follows:

$$\lambda_P \cdot P \oplus \lambda_C \cdot C = \lambda_K \cdot K$$

where $\lambda_i$ are bit masks and $\lambda_X \cdot X$ is a scalar product of $\lambda_X$ and $X$. Stated differently, $\lambda_X \cdot X$ denotes the XOR sum (parity) of the bits of $X$ masked (selected) by $\lambda_X$.

Let $p$ denote the probability that $\lambda_K \cdot K = 0$. If $P$, $C$ and $K$ are random variables, then $p = 1/2$. However, for some block ciphers, there may exist cases where $p = 1/2 + q$ in which $q$ is known as the bias, and $0 < |q| \leq 1/2$. It may be possible to exploit this relationship to recover round subkey bits.

The tool used to detect bias is the Linear Approximation Table (LAT) for an $m \times n$ S-box. Let $\lambda_\alpha$ and $\lambda_\beta$ denote the input and output masks of an S-box, respectively. The LAT contains $2^m$ rows that denote all possible input masks $\lambda_\alpha$ and $2^n$ columns that denote all possible output masks $\lambda_\beta$. An entry in the LAT represents the number of occurrences that an input parity masked by $\lambda_\alpha$ equals an output parity masked by $\lambda_\beta$, minus half the number of possible inputs. The bias for the corresponding event, denoted $\lambda_\alpha \rightarrow \lambda_\beta$, can be calculated from the LAT as follows

$$q = \frac{\text{LAT}[\lambda_\alpha][\lambda_\beta]}{2^m}$$

where the probability is $p = 1/2 + q$. An algorithm to calculate the entries in a LAT is given in Algorithm 2.2.

In the distinguishing phase of the attack, an $\hat{R}$-round linear characteristic is constructed by concatenating $\hat{R}$ 1-round characteristics. The 1-round characteristic is built by approximating the parities for selected S-boxes in the round. If this $\hat{R}$-round characteristic involves $m_l$ linearly active\footnote{A linearly active S-box is an S-box that is approximated in the characteristic, that is, its input and output masks are nonzero.} S-boxes, then the total
begin
  for $\lambda_\alpha = 0$ to $2^m - 1$ do
    for $\lambda_\beta = 0$ to $2^n - 1$ do
      for $X = 0$ to $2^m - 1$ do
        if $\lambda_\alpha \cdot X = \lambda_\beta \cdot s[X]$ then
          Increment LAT[$\lambda_\alpha$][$\lambda_\beta$];
        end
      end
    end
  end
  Subtract $2^m/2$ from each entry in LAT;
end

**Algorithm 2.2:** Algorithm to calculate the Linear Approximation Table (LAT)

bias is estimated using the Piling-Up lemma \[132\]:

$$q = 2^{m_1 - 1} \prod_{i=0}^{m_1-1} \hat{q}_i$$

where $\hat{q}_i$ denotes the bias for S-box $i$.

In a basic $R$-round key recovery attack using an $\hat{R}$-round distinguisher where $\hat{R} = R - 1$, an additional round is appended at the end of the $\hat{R}$-round characteristic. A collection of plaintext and ciphertext pairs encrypted by the $R$-round block cipher under the same secret key is obtained. The affected word of the subkey in the $R$-th round is guessed. For each guess, the affected ciphertext words are partially decrypted to obtain the output of the affected S-boxes in round $\hat{R}$. The XOR sum of selected bits (determined during the distinguishing phase) of the resulting output and plaintext are calculated. If the XOR sum equals zero, then the counter corresponding to the guessed subkey value is incremented by one. The attack is repeated with other plaintext and ciphertext pairs. In the end, the subkey word that is counted the most is taken as the correct value of the subkey word.

According to Matsui, the data complexity of the attack is proportional to $q^{-2}$ \[132\]. In order to increase the success rate of the attack, the number of known plaintexts required is a multiple of $q^{-2}$, that is $c \cdot q^{-2}$ where $c > 1$.

---

3Note that the attacker only collects the plaintext and ciphertext pairs but is not allowed to choose particular plaintexts to encrypt.
In general, the time complexity of the attack relates to the number of partial decryptions and the memory complexity refers to the amount of space required to store the counters.

In attacking the full 16-round DES, Matsui used two 14-round expressions with $2^{43}$ known plaintexts [131]. The expressions allow an attacker to obtain 26 bits of the key bits while the remaining 30 bits can be recovered using exhaustive key search.

### 2.3.2 Differential Cryptanalysis

Differential cryptanalysis [23, 24] is a chosen plaintext attack that exploits the existence of a linear relationship between the encryption of two plaintexts using the same secret master key. Given a pair of plaintexts $X^{0(0)}$ and $X^{0(1)} = X^{0(0)} \oplus \Delta_P$, the ciphertexts after $R$ rounds are respectively $X^{R(0)}$ and $X^{R(1)} = X^{R(0)} \oplus \Delta_X$ with high probability. The expected difference at this round is used in a key recovery attack to identify possible subkey bits.

The tool used to calculate the probability for this attack is the Difference Distribution Table (DDT) for an $m \times n$ S-box. Let $\Delta_\alpha$ and $\Delta_\beta$ denote the input and output differences of an S-box, respectively. The DDT contains $2^m$ rows that denote all possible input differences $\Delta_\alpha$ and $2^n$ columns that denote all possible output differences $\Delta_\beta$. An entry in the DDT represents the number of times that an input difference $\Delta_\alpha$ causes an output difference $\Delta_\beta$ to occur. This event is denoted by $\Delta_\alpha \rightarrow \Delta_\beta$. The probability of the occurrence of this event can be calculated from the DDT as follows

$$p = \frac{\text{DDT}[\Delta_\alpha][\Delta_\beta]}{2^m}.$$

An algorithm to calculate the entries in a DDT is given in Algorithm 2.3. The DDT tables for the block ciphers analyzed in this thesis are given in Appendix B.

In the distinguishing phase of the attack, an $\hat{R}$-round differential characteristic is constructed by concatenating $\hat{R}$ 1-round characteristics. The 1-round characteristic is built by predicting the differences for selected S-boxes in the round. Let $\Delta_P$ denote the plaintext difference and $\Delta_X$ denote the output difference after $\hat{R}$ rounds. If this $\hat{R}$-round characteristic involves $m_d$ differentially
Algorithm 2.3: Algorithm to calculate the Difference Distribution Table (DDT)

\[
\begin{align*}
\text{begin} & \quad \text{for } \Delta_\alpha = 0 \text{ to } 2^m - 1 \text{ do} \\
& \quad \quad \text{for } X = 0 \text{ to } 2^m - 1 \text{ do} \\
& \quad \quad \quad \Delta_\beta = s[X] \oplus s[X \oplus \Delta_\alpha]; \\
& \quad \quad \text{Increment DDT}[\Delta_\alpha][\Delta_\beta]; \\
& \quad \text{end} \\
& \text{end} \\
\text{end}
\end{align*}
\]

active S-boxes, then the total probability is:

\[
p = \prod_{i=0}^{m_d-1} \hat{p}_i
\]

where \(\hat{p}_i\) denotes the probability for S-box \(i\) with the assumption that the round subkeys are independent. A set of different \(R\)-round characteristics but with the same input and output differences can be collected as a differential \(^{12}\). In general, a differential has a higher probability than that of a characteristic.

In a basic \(R\)-round key recovery attack using an \(\hat{R}\)-round distinguisher where \(\hat{R} = R - 1\), an additional round is appended at the end of the \(\hat{R}\)-round characteristic. Plaintext pairs that have the difference \(\Delta_P\) are encrypted and the ciphertext pairs after \(R\) rounds are obtained. The affected word of the subkey in the \(R\)-th round is guessed. For each guess, the affected ciphertext pairs are partially decrypted to obtain the output difference of the affected S-boxes in round \(\hat{R}\). If the resulting difference matches the difference predicted by the characteristic (\(\Delta_X\)), then the guessed subkey value is a possible correct word of the subkey. Note that there may be more than one possible correct subkey values that causes this event to occur. Therefore, the attack is repeated with different pairs of plaintext blocks. In the end, the subkey word that is counted the most is taken as the correct value of the subkey word.

In general, similar to linear cryptanalysis, the data complexity of the attack is proportional to \(p^{-1}\). The amount of chosen plaintexts required is a multiple of \(p^{-1}\), that is \(c \cdot p^{-1}\) where \(c > 1\). The time complexity of the attack is the number of partial decryptions and the memory complexity is the amount of space needed.

\(^{4}\)A differentially active S-box is an S-box that is approximated in the characteristic, that is, its input and output differences are nonzero.
2.3. Existing Cryptanalysis Techniques

to store the counters.

2.3.3 Truncated and Higher-Order Differentials

Differential cryptanalysis deals with differences for the entire block. In contrast, a truncated differential \[108\] considers differences for only a subset of the input and output blocks. In general, this technique yields better characteristics than traditional differential cryptanalysis. Furthermore, differential cryptanalysis uses a differential of order one (first-order differential). This method can be generalized and extended in a higher-order differential \[108, 119\] by using two pairs (a quartet/second order differentials), two quartets (third order) and so on (higher order). Note that a block cipher that is immune to conventional differential cryptanalysis may be susceptible to truncated or higher-order differential cryptanalysis.

2.3.4 Impossible Differentials

Differential cryptanalysis deals with the occurrences of high probability events. On the other hand, an impossible differential \[13\] deals with the occurrences of zero-probability events. The technique is a sieving attack which eliminates wrong subkey guesses that lead to contradictions. Eventually, only the right possible subkeys remain. A miss-in-the-middle technique is used in which two input differences are propagated to the middle of a cipher, one from the top and one from the bottom. If the resulting output differences in the middle are distinct, then a contradiction occurs. Subkey guesses that cause such an event to happen are discarded. The attack can also be used with low probability differentials and conditional characteristics (or differentials), and can be combined with linear cryptanalysis.

2.3.5 Boomerang and Rectangle Attacks

The boomerang attack, introduced by Wagner, is an adaptive chosen plaintext and ciphertext attack based on differential cryptanalysis \[173\]. The attack treats a cipher as the composition of two sub-ciphers \(E = E_1 \circ E_0\). A quartet structure is then created in the middle using two differential characteristics which hold with high probability halfway through the cipher, one covering the first half and the other covering the second half.
Figure 2.3: A boomerang distinguisher

Given a pair of plaintexts $P_1$ and $P_2 = P_1 \oplus \Delta$, one calculates their corresponding ciphertexts $C_1 = E(P_1)$ and $C_2 = E(P_2)$. The difference $\Delta$ is chosen so that the difference $\Delta^* = E_0(P_1) \oplus E_0(P_2)$ is obtained halfway through the cipher with high probability. From the ciphertexts, one creates another pair $C_3 = C_1 \oplus \nabla$ and $C_4 = C_2 \oplus \nabla$ and decrypt them to obtain $P_3 = E^{-1}(C_3)$ and $P_4 = E^{-1}(C_4)$. The difference $\nabla$ is chosen so that the difference $\nabla^*$ is expected to happen in the decryption direction through the lower half of the cipher, i.e. $\nabla^* = E^{-1}_1(C_1) \oplus E^{-1}_1(C_3) = E^{-1}_1(C_2) \oplus E^{-1}_1(C_4)$. A right quartet is detected if the difference $\Delta^*$ is created in the middle of the cipher and it propagates to the difference $\Delta = P_3 \oplus P_4$ in the second pair of plaintexts. This is depicted in Figure 2.3. Note the direction of the differential indicated by the arrow. Subkeys in the outer rounds of the distinguisher are guessed and verified with the distinguisher in a key recovery attack.

The boomerang attack is improved into a chosen plaintext attack by Kelsey, Kohno and Schneier [99]. The technique, called amplified boomerang, involves the encryption of many plaintext pairs so that the difference expected in the middle of the cipher appears by chance. The attack setting is very similar to the boomerang attack. Recall from above that in a basic boomerang attack, two
plaintexts are encrypted and then, based on the resulting ciphertexts, two new ciphertexts are generated and decrypted. So, two encryptions and two decryptions are performed. In an amplified boomerang attack, four encryptions are performed. The rectangle attack \cite{14,16} further enhances the amplified boomerang by using differentials instead of characteristics. This method increases the likelihood of identifying the right quartets.

The boomerang, amplified boomerang and rectangle attacks show that a differential-style attack can be successfully applied to a cipher that claims to be immune to conventional differential cryptanalysis. Since these boomerang type attacks combine short differentials with high probability, they are generally more effective than using a single but long differential.

2.3.6 Integral Attack

The integral attack \cite{112} is a chosen plaintext attack that was initially developed to attack the block cipher Square \cite{59}. The attack exploits the behaviour that the XOR sum of a structure of texts can be determined after several iterations of the round function. In order for this event to happen, the structure of plaintexts has to be in a particular format. The format of this structure is explained later in this section. The point before the XOR sum of the texts can no longer be determined is used in a key recovery attack to identify possible round subkey bit values. The attack is particularly suited for block ciphers where all transformations operate on words (word-based block ciphers).

Integral Properties

An integral structure is a collection of \( w \) text blocks \([W^{r(0)}, W^{r(1)}, \ldots, W^{r(w-1)}]\) where \(W^{r(l)} = (W_0^{r(l)}, W_1^{r(l)}, \ldots, W_{m-1}^{r(l)})\) denotes the \( l \)-th text block of the structure in round \( r \), formed from the concatenation of \( m \) \( b \)-bit words \( W_i^{r(l)} \). The \( i \)-th word of the structure, i.e. \( W_i^{r(l)} \) (for \( l = 0, 1, \ldots, w - 1 \)), has one of the following integral properties:

- Property A (All), if all possible \( b \)-bit texts exist in the word.
- Property C (Constant) if all texts in the word are identical.
- Property B (Balanced\(^5\)) if the XOR sum of all texts in the word can be

\(^5\)Note that the definition of balanced here is not the same as the definition of a balanced
Table 2.1: Example of text values for a structure of sixteen text blocks which has the property ACCB

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>A</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

determined in advance, i.e. $\bigoplus_{i=0}^{w-1} W_i^{(l)} = d$ where $d$ is a predetermined $b$-bit value. In this thesis, the focus is only on the value $d = 0$.

- Property ? (Unknown) if the XOR sum of all texts in the word cannot be determined in advance.

Note that a word of a structure that has the property A or C also has the property B. An example of a structure of sixteen text blocks (in hexadecimal) where each text block is formed from the concatenation of four 4-bit words, is given in Table 2.1. In this example, the structure has the property ACCB.

**Tracing the Words using Properties**

The words of the structure will evolve after each application of the transformations in the round function. Depending on the exact values of the texts, the changed words may have different properties. In the remainder of this section, only word-based bijective transformations are considered. In particular, the linear transformation is assumed to consist of at least one of these operations: XOR, boolean function.
rotation, shift. For the nonlinear transformation, it is assumed to be composed of bijective S-boxes.

A linear transformation normally preserves the balance of a structure and retains the properties $A$ and $C$ for particular words. In particular, a linear key mixing transformation which uses the XOR operation does not have any affect on the integral properties. If the input word to a nonlinear S-box has the property $A$ or $C$, then its output word will always have the same property as its input. If, however, the input word to an S-box has the property $B$, then the output word will normally evolve to have the property $?$. This is because in a real-world attack, one does not know the exact values of the structure at the intermediate stages of the cipher due to the existence of the key mixing transformation. Therefore, even though the XOR sum of the texts in a word which have the integral property $B$ before the S-box is known, the XOR sum after the S-box cannot be known for certain. Example 2.1 illustrates this scenario where a structure which has the integral property $ACCB$ evolve into a structure with the property $ACC?$.

Example 2.1 Let $\begin{bmatrix} 0A41, 1A42, 2A43, 3A40, 8A40, 9A40, AA40, BA40, 4A40, 5A40, 6A40, 7A40, CA40, DA40, EA40, FA40 \end{bmatrix}$ represent a structure of sixteen text blocks, where each text block is formed from the concatenation of four 4-bit words. This structure has the integral property $ACCB$. Assume that this structure is used as input to a nonlinear transformation $S$ which is composed of the application of four identical S-boxes $s$, and the following output structure is obtained:

$\begin{bmatrix} E14A, A142, 214C, C14E, 514E, 914E, 114E, B14E, 414E, 814E, F14E, D14E, 314E, 714E, 014E, 614E \end{bmatrix}$. The integral property of this output structure of $S$ is $ACC?$. It can be observed that in the first word of each text block, one still has all possible 4-bit values. In the second and third words, one still has a single value repeated sixteen times. In the fourth word, the XOR sum of the texts is nonzero and the property of this word is denoted by $?$.

Distinguishing Phase

In order to build a basic distinguisher for the integral attack, the plaintext structure is chosen in a particular format such that the words have the properties $A$ and $C$ only. The properties of these words are traced as the words propagate through the cipher. After $\hat{R}$ rounds, the values of the structure words will eventually have changed and all or some of the structure words might have the property $?$. If all words have the property $?$, then the balance of the structure
can no longer be determined. This event normally occurs after a nonlinear transformation. If this event occurs after the nonlinear transformation in round $\hat{R}$, then at least one S-box in this round, for instance the $i$-th S-box, satisfies the following equation.

$$\bigoplus_{i=0}^{w-1} Y_i^{\hat{R}(l)} = \bigoplus_{i=0}^{w-1} s^{-1}(Z_i^{\hat{R}(l)}) = 0. \quad (2.1)$$

In essence, if one knows that the input words to the $i$-th S-box in round $\hat{R}$ has property B, then one can compute backwards from round $R$ where $\hat{R} = R - 1$ and obtains the zero-sum at the input of this particular S-box. Note that since the balance of the structure can only be determined before any of the S-boxes in round $\hat{R}$, one will have an $\hat{R}$-round integral distinguisher. This distinguishing phase is outlined in Algorithm 2.4.

```
begin
    Input: Specific block cipher;
    Generate plaintext blocks that have properties A and C;
    Insert plaintext blocks into block cipher;
    Trace the propagation of properties until round $\hat{R}$;
    if all words have property ? in round $\hat{R}$ then
        Output: $\hat{R}$-round distinguisher;
    else
        $\hat{R}$-round distinguisher unavailable;
    end
end
```

**Algorithm 2.4**: Algorithm for constructing an integral distinguisher

**Key Recovery Phase**

In a basic $R$-round key recovery attack using an $\hat{R}$-round distinguisher where $\hat{R} = R - 1$, an additional round is appended at the end of the $\hat{R}$-round integral distinguisher. A plaintext structure that follows the format of the distinguisher is encrypted and the ciphertext structure after $R$ rounds is obtained. The affected word of the subkey in the $R$-th round is guessed to obtain the texts for the word $Z_i^{\hat{R}(l)}$. Then, the affected ciphertext word is partially decrypted using the guessed subkey word. If Equation 2.1 does not hold, then the guessed subkey value is wrong and is discarded.\(^6\) The guess is performed multiple times for all possible

\(^6\)Note that this elimination technique can only be used if Equation 2.1 holds with probability one.
values of the affected word of the subkey. The remaining guessed value which is not discarded, is the correct subkey. Note that there may be more than one possible subkey value that causes this event to occur. Therefore, the attack is repeated with different sets of plaintext blocks until only a single subkey remains. This key recovery phase is outlined in Algorithm 2.5 and an example of a key recovery attack is given in Example 2.2.

Algorithm 2.5: Algorithm for $R$-round key recovery phase in an integral attack

Example 2.2 Suppose that the $i$-th 4-bit word of a structure of 16 ciphertexts can be written as $C_i^{(l)} = Z^{r(l)}_i \oplus K^{r(l)}_i$ and one needs to recover the 4-bit subkey $K^{r(l)}_i$. From Equation 2.7 it follows that the ciphertext word can be rewritten as $Y^{r(l)}_i = s^{-1}(C^{(l)}_i \oplus K^{r(l)}_i)$. Therefore, for each 4-bit guess $K^{r(l)}_i$, one needs to obtain the value of $Y^{r(l)}_i$ for $l = 0, 1, \ldots, 15$ and check whether Equation 2.7 holds. If it does not hold, then discard the guessed value. The single remaining value is the correct subkey.

Application to the AES

Figure 2.4 illustrates a 3-round integral distinguisher for the AES. The transformation $S$ is nonlinear, while $L_0$ and $L_1$ are linear. All these transformations are discussed in detail in Section 2.4.1. In this case, the structure of the text blocks is divided into sixteen 8-bit words. The plaintext structure is chosen such
that only one word has the property A and the remaining words have the property C. After the first round, four words of the structure have the property A while the remaining twelve words have the property C. After the second round, all words of the structure have the property A. After the third round, all words have the property B. This property is retained until before the next nonlinear transformation S in the fourth round.

In a basic key recovery attack, every subkey byte in the fourth round is guessed and the corresponding word of the ciphertext structure is decrypted using the guessed subkey byte. If the word of the resulting decryption has the property B, then the guessed byte is a possible correct key.

### Figure 2.4: A 3-round integral distinguisher for the AES

The integral attack works well on ciphers with a word-based structure and is independent of the choice of the bijective S-Box. One limitation of this attack is that in most cases, the basic distinguisher can only be constructed to cover small number of rounds. The integral attack is similar to the multiset [34] and the saturation attack [129].
2.3.7 Slide Attack

The slide attack is a known- (or sometimes chosen-) plaintext attack introduced by Biryukov and Wagner [35]. In certain circumstances, the attack is independent of the number of rounds and can be applied to block ciphers that iterate a weak round function and have a periodic key scheduling algorithm. In general, the attack works by sliding the encryption of two plaintexts $P^{(0)}$ and $P^{(1)}$, so that the second plaintext is equal to the application of the first round function to the first plaintext as follows:

$$P^{(1)} = F_{K^0}(P^{(0)}).$$

The attack assumes that all round functions are identical, that is $F_{K^i} = F_{K^{i+1}}$ for $i \geq 1$. Therefore, after $r$ rounds, the ciphertext $C^{(1)}$ of the second encryption is equal to the application of the last round function to the first ciphertext $C^{(0)}$ as follows:

$$C^{(1)} = F_{K^{r-1}}(C^{(0)}).$$

The pair $(P^{(0)}, C^{(0)}), (P^{(1)}, C^{(1)})$ is called a slid pair. The last round subkey is recovered by guessing the value of $K^{r-1}$ so that the above equation holds.

The slide attack is later improved by its designers with the introduction of the complementation slide and slide with-a-twist techniques [36]. The first technique is useful in amplifying the self-similarity in iterated ciphers. The second technique permits sliding encryption with decryption. The attack is further improved by Biham, Dunkelman and Keller [22] by proposing a more efficient method for finding slid pairs. This technique reduces the time complexity at the expense of data complexity.

2.3.8 Related-Key Attacks

Related-key attacks allow the attacker to manipulate the key scheduling algorithms of block ciphers [11,100,101,106]. The attacks also permit the attacker to choose a relationship between two or more different master keys but not the actual value of the keys. Additionally, the attacker may also choose the relationship between the plaintexts. In one variant of this kind of attack, the plaintexts are encrypted under this set of related keys. The subkeys are then recovered in a differential-like manner. The existence of related-keys can also reduce the amount of searched key space [21,106].

Related-key attacks are often differentiated from attacks using a single master
key since the latter is more practical than the former. Although not very practical in the real-world, related-key attacks have been shown to be useful in attacking the full version of block ciphers such as KASUMI \cite{18}, SHACAL-1 \cite{21,70}, 192-bit key AES \cite{31,32} and the 256-bit key AES \cite{33}. In terms of time complexity, Biryukov et al. manage to demonstrate a practical attack on variants of the AES with up to 10 rounds \cite{30}.

### 2.3.9 Algebraic Cryptanalysis

Algebraic attacks are known plaintext attacks that exploit the algebraic structure within a cipher \cite{55,56}. The attacks typically involve two phases: deriving a system of equations and then solving the system. In the first phase, the attacker builds a system of equations (linear and nonlinear) that involve the plaintext, subkey, ciphertext and unknown intermediate texts of the encryption. In the second phase, these equations are solved in order to recover the key.

The system of equations is typically very large and the main challenge is to find an efficient method to solve these equations. One basic method is linearization where quadratic expressions are replaced with new terms so that the equations become linear. Other proposed methods are relinearization \cite{105}, XL \cite{52}, XSL \cite{55}, Gröbner basis \cite{46} and the Raddum-Semaev technique \cite{170}. In particular, in the Gröbner basis method, a set of polynomials is converted into another set of polynomials called a Gröbner basis. This new set has certain properties that makes the set easier to solve than the original set. The solutions found for this new set can be converted to the solutions for the original set. The reader is referred to Becker and Weispfenning \cite{10} for a more detailed treatment of Gröbner basis. This method will be used in Chapter 4.

The AES has been the main target of analysis since it is considered to possess rich algebraic properties \cite{28,40,46,77,136}. Among others, the DES \cite{53} and SMS4 \cite{91} have also been investigated using algebraic approaches. However, to date, none of these approaches manage to demonstrate a successful key recovery attack. In Chapter 4, a block-cipher-based stream cipher is analyzed using an algebraic approach.
2.4 Analyzed Block Ciphers

This section describes the block ciphers which will be analyzed throughout this thesis. All block ciphers analyzed use an SPN network, except for SMS4, which is an unbalanced Feistel cipher.

2.4.1 AES

As mentioned at the beginning of this chapter, the Advanced Encryption Standard (AES) \[144\] is a standard block cipher adopted by the NIST of the United States (US). In 1997, the NIST called for an open evaluation process to select the AES to replace the then 20-year old DES \[141\]. Out of the 15 block cipher candidates submitted to the process, Rijndael \[62\] was eventually selected to become the AES in 2000. It has since become a global de facto standard. In 2003, the US government broadened the scope of the AES to include protection of classified information up to SECRET and TOP SECRET levels \[50\].

The AES accepts a 128-bit plaintext block \( P \), and a master key \( K \) with allowable sizes of 128, 192 and 256 bits. The master key is used as input to the key scheduling algorithm to produce a set of \((R+1)\) 128-bit round subkeys. The ciphertext block \( C \) is produced after the round function is applied \( R \) times. The number of rounds \( R \) is 10 if the master key size is 128 bits, 12 if 192 bits and 14 if 256 bits.

**Encryption Algorithm**

The plaintext block \( P = (X_0^0, X_1^0, \ldots, X_{15}^0) \) is formed from the concatenation of sixteen 8-bit words \( X_i^0 \). Let \( K^r \) denote the 128-bit subkey in round \( r \) derived from the master key \( K \). The derivation of these subkeys is explained later in this section. The round function is composed of a nonlinear transformation \( S \), linear transformations \( L_0 \) and \( L_1 \), and a key mixing transformation. These transformations are defined in detail later. The encryption algorithm of the AES can be expressed by the following equations:

\[
X^1 = P \oplus K^0 \\
X^{r+1} = L_1(L_0(S(X^r))) \oplus K^r, \quad r = 1, 2, \ldots, R - 1 \\
C = L_0(S(X^R)) \oplus K^R
\]
where \( R \in \{10, 12, 14\} \). It can be observed that there is an additional key mixing transformation before the first round, and that the transformation \( L_1 \) is omitted in the last round. The round function of the AES is depicted in Figure 2.5.

The nonlinear transformation \( S \), also known as \texttt{SubBytes}, accepts the block \( X^r = (X_0^r, X_1^r, \ldots, X_{15}^r) \) as input which is composed of sixteen 8-bit words \( X_i^r \). The application of \( S \) to \( X^r \) consists of the application of a single 8 \( \times \) 8 S-box \( s \) to \( X^r \) sixteen times in parallel to produce the output block \( Z^r \). The transformation \( S \) is denoted as follows.

\[
Z^r = (Z_0^r, Z_1^r, \ldots, Z_{15}^r) = S(X^r) = (s(X_0^r), s(X_1^r), \ldots, s(X_{15}^r)).
\]

The linear transformation \( L_0 \), also known as \texttt{ShiftRows}, produces the block \( \hat{Y}^r = (\hat{Y}_0^r, \hat{Y}_1^r, \ldots, \hat{Y}_{15}^r) \) as output, which is formed from the concatenation of sixteen 8-bit words \( \hat{Y}_i^r \). The application of the linear transformation \( L_0 \) to \( Z^r \) is described as follows.

\[
\hat{Y}_i^r = \begin{cases} 
Z_i^r & \text{for } i = 0, 4, 8, 12 \\
Z_{i+4}^r & \text{for } i = 1, 5, 9, 13 \\
Z_{i+8}^r & \text{for } i = 2, 6, 10, 14 \\
Z_{i+12}^r & \text{for } i = 3, 7, 11, 15
\end{cases}
\]

The linear transformation \( L_1 \), also known as \texttt{MixColumns}, produces the block \( Y^r = (Y_0^r, Y_1^r, \ldots, Y_{15}^r) \) as output, which is formed from the concatenation of sixteen 8-bit words \( Y_i^r \). The application of the linear transformation \( L_1 \) to \( \hat{Y}^r \) is
where multiplications are performed in a finite field.

The above linear transformations \( L_0 \) and \( L_1 \) can also be depicted as a 16 \( \times \) 16 matrix given as follows.

\[
\begin{bmatrix}
Y_0^r \\
Y_1^r \\
Y_2^r \\
Y_3^r \\
Y_4^r \\
Y_5^r \\
Y_6^r \\
Y_7^r \\
Y_8^r \\
Y_9^r \\
Y_{10}^r \\
Y_{11}^r \\
Y_{12}^r \\
Y_{13}^r \\
Y_{14}^r \\
Y_{15}^r
\end{bmatrix} =
\begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\
0 & 1 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 0 \\
\end{bmatrix}
= 
\begin{bmatrix}
Z_0^r \\
Z_1^r \\
Z_2^r \\
Z_3^r \\
Z_4^r \\
Z_5^r \\
Z_6^r \\
Z_7^r \\
Z_8^r \\
Z_9^r \\
Z_{10}^r \\
Z_{11}^r \\
Z_{12}^r \\
Z_{13}^r \\
Z_{14}^r \\
Z_{15}^r
\end{bmatrix}
\]

The key mixing transformation, also known as AddRoundKey, accepts the block \( Y^r \) as input and produces the output block of round \( r \) as follows \( X_{i+1}^r = Y_i^r \oplus K_i^r \), for \( i = 0, 1, \ldots, 15 \).

**Key Scheduling Algorithm**

The master key \( K = (K_0, K_1, \ldots, K_{m_k-1}) \) is formed from the concatenation of \( m_k \) 8-bit words. The value \( m_k \) is 16 for 128-bit master key, \( m_k = 24 \) for 192-bit key and \( m_k = 32 \) for 256-bit key. There are two variants of the key scheduling algorithm: one for both 128-bit and 192-bit master keys, and one for 256-bit master key. In this thesis, the focus is only on the first variant of the key scheduling algorithm (128- and 192-bit master keys).

The key scheduling algorithm is described as follows. First, the master key \( K \) is put into a two-dimensional array \( K[i][j] \) consisting of 4 rows and \( m_k/4 \)
columns. The setup is performed as follows: \( \hat{K}[i \mod 4][i/4] = K_i \) for \( i = 0, 1, \ldots, m_k/4 - 1 \). Next, an array \( G[\cdot][\cdot] \) consisting of 4 rows and 4\((R+1)\) columns is initialized. The key array \( \hat{K} \) is expanded into the array \( G \) by Algorithm 2.6. Then, byte \( i \) of the 128-bit subkey in round \( r \), denoted \( K^r_i \), is derived as follows.

\[
K^r_i = G[i \mod 4][4r + [i/4]] \quad i = 0, 1, \ldots, 15.
\]

begin
| Input: Master key \( \hat{K} \) and array \( G[4][4(R+1)] \); |
| for \( i = 0 \) to 3 do |
| for \( j = 0 \) to \( m_k/4 - 1 \) do |
| \( G[i][j] = \hat{K}[i][j] \); |
| end |
| end |
| for \( j = m_k/4 \) to \( 4(R+1) - 1 \) do |
| if \( j \mod 4 \) equals 0 then |
| \( G[0][j] = G[0][j - 4] \oplus s(G[1][j-1]) \oplus RC^{4j/m_k} \); |
| for \( i = 1 \) to 3 do |
| \( G[i][j] = W[i][j - m_k/4] \oplus s(G[(i+1) \mod 4][j-1]) \); |
| end |
| else |
| for \( i = 0 \) to 3 do |
| \( G[i][j] = G[i][j - m_k/4] \oplus G[i][j-1] \); |
| end |
| end |
| Output: Array \( G \); |
end

Algorithm 2.6: Key Scheduling Algorithm for the AES.

Previous Cryptanalysis

As a standard, the AES block cipher has been subjected to great public scrutiny, resulting in a number of attacks. Some of these attacks are given in Table 2.2. The table is divided into three main rows. The first main row contains attacks for the AES-128, the second main row lists attacks for AES-192 and the last main row shows attacks for the AES-256. Previous cryptanalysis on the AES can be categorized into single- and related- (or multiple-) key attacks.

In the single-key scenario, various attacks have been mounted on the AES-
128 reduced to five \[25\], six \[25, 44, 76\] and seven \[80, 127, 184\] rounds. For AES-192, there are attacks on seven \[63, 76, 80, 127, 128, 153, 184\] and eight \[76\] rounds. For AES-256, there are attacks on seven \[63, 127, 153, 184\] and eight \[63, 76, 127, 184\] rounds. The partial sums technique \[76\], which is based on the integral attacks, is one of the best attacks on all key-size variants of reduced-round AES. The attacks manage to penetrate the AES-128 reduced to seven rounds, and both AES-192 and AES-256 reduced to eight rounds. Other best attacks include impossible differential \[127\] and meet-in-the-middle \[63\] attacks. However, progress in advancing the number of attacked rounds in the single-key scenario is slow. No other attacks have been successful in attacking more rounds than the partial sums technique, which was introduced in 2000.

In the related-key scenario, there has been significant progress in the cryptanalysis of both the AES-192 and AES-256. This is due to the slower diffusion property in the key scheduling algorithms for these variants compared to the AES-128. For the 12-round AES-192, related-key type attacks manage to penetrate seven \[90\], eight \[20, 90, 186\], nine \[19\] and ten \[103\] rounds. Similarly, for the 14-round AES-256, related-key attacks can be launched on nine \[30, 76, 78\] and ten \[19, 30, 78, 103\] rounds. Recently, Biryukov et al. demonstrated attacks on the full AES-192 and AES-256 using related-key amplified boomerang and boomerang, respectively \[31, 32\]. Both attacks use only four related keys and are the best existing attacks on AES-192 and AES-256.

### 2.4.2 ARIA

The block cipher ARIA was first introduced at an annual security conference in Korea in 2003 \[145\]. This variant, which is called version 0.8, uses two different S-boxes and 10, 12 and 14 rounds (depending on the size of the master key). Later that year, the cipher was upgraded to version 0.9 with the introduction of four different S-boxes \[116\]. In 2004, the National Security Research Institute (NSRI) of Korea announced ARIA version 1.0 on its website. This variant uses 12, 14 and 16 rounds and a modified key scheduling algorithm. In December 2004, the Korean Agency for Technology and Standards (ATS) adopted ARIA version 1.0 as a standard block cipher \[146\]. In this thesis, focus is on this latest

---

\[7\] Here, the best attacks refer to attacks that manage to penetrate the most number of block cipher rounds.

\[8\] These attacks improve the original 6-round attack of Daemen and Rijmen \[61\] and Daemen, Knudsen and Rijmen \[59\].
version of ARIA.

ARIA accepts a 128-bit plaintext block $P$ and a master key $K$ with allowable sizes of 128, 192 and 256 bits. The key scheduling algorithm uses the master key as input to generate a set of $(R+1)$ 128-bit round subkeys. The ciphertext block $C$ is produced after the application of the round function $R$ times. The number of rounds $R$ is 12 if the master key size is 128 bits, 14 if 192 bits and 16 if 256 bits.

The round function is composed of a key mixing transformation, a nonlinear transformation $S_r$, and a linear transformation $L$. In the last round, the transfor-
2.4. Analyzed Block Ciphers

Figure 2.6: Round function and nonlinear transformations of ARIA in Round $r$

ARIA uses two different nonlinear transformations denoted $S_0$ and $S_1$. Let $Y^r$ denote the output block of the key mixing transformation, i.e. $Y^r = X^r \oplus K^r$. The nonlinear transformation $S_r$ consists of four $8 \times 8$ S-boxes denoted $s_1$, $s_2$, $s_1^{-1}$ and $s_2^{-1}$. The positions of these S-boxes are depicted in Figure 2.6. The nonlinear transformation processes the input block $Y^r$ to produce the output block $Z^r$.

Let $X^{r+1} = (X_0^{r+1}, X_1^{r+1}, \ldots, X_{15}^{r+1})$ denote the output block of $L$ composed of 16 8-bit words $X_i^{r+1}$. The transformation $L$ of ARIA uses the block $Z^r$ as input to produce the output block $X^{r+1}$, i.e. $X^{r+1} = L(Z^r)$. Specifically, this is
expressed as follows:

\[
\begin{align*}
X_0^{r+1} &= Z_3^r + Z_4^r + Z_6^r + Z_8^r + Z_9^r + Z_{13}^r + Z_{14}^r \\
X_1^{r+1} &= Z_2^r + Z_5^r + Z_7^r + Z_8^r + Z_9^r + Z_{12}^r + Z_{15}^r \\
X_2^{r+1} &= Z_1^r + Z_4^r + Z_6^r + Z_{10}^r + Z_{11}^r + Z_{12}^r + Z_{15}^r \\
X_3^{r+1} &= Z_0^r + Z_5^r + Z_7^r + Z_{10}^r + Z_{11}^r + Z_{13}^r + Z_{14}^r \\
X_4^{r+1} &= Z_0^r + Z_2^r + Z_5^r + Z_8^r + Z_{11}^r + Z_{14}^r + Z_{15}^r \\
X_5^{r+1} &= Z_1^r + Z_3^r + Z_4^r + Z_9^r + Z_{10}^r + Z_{14}^r + Z_{15}^r \\
X_6^{r+1} &= Z_0^r + Z_2^r + Z_7^r + Z_9^r + Z_{10}^r + Z_{12}^r + Z_{15}^r \\
X_7^{r+1} &= Z_1^r + Z_3^r + Z_6^r + Z_8^r + Z_{11}^r + Z_{12}^r + Z_{13}^r \\
X_8^{r+1} &= Z_0^r + Z_1^r + Z_4^r + Z_7^r + Z_{10}^r + Z_{13}^r + Z_{15}^r \\
X_9^{r+1} &= Z_0^r + Z_1^r + Z_5^r + Z_6^r + Z_{11}^r + Z_{12}^r + Z_{14}^r \\
X_{10}^{r+1} &= Z_2^r + Z_3^r + Z_5^r + Z_6^r + Z_8^r + Z_{13}^r + Z_{14}^r \\
X_{11}^{r+1} &= Z_2^r + Z_3^r + Z_4^r + Z_7^r + Z_9^r + Z_{12}^r + Z_{14}^r \\
X_{12}^{r+1} &= Z_1^r + Z_2^r + Z_6^r + Z_7^r + Z_9^r + Z_{11}^r + Z_{12}^r \\
X_{13}^{r+1} &= Z_0^r + Z_3^r + Z_6^r + Z_7^r + Z_8^r + Z_{10}^r + Z_{13}^r \\
X_{14}^{r+1} &= Z_0^r + Z_3^r + Z_4^r + Z_5^r + Z_6^r + Z_{11}^r + Z_{14}^r \\
X_{15}^{r+1} &= Z_1^r + Z_2^r + Z_4^r + Z_5^r + Z_8^r + Z_{10}^r + Z_{15}^r.
\end{align*}
\]

The above transformation \( L \) can also be depicted as a 16\( \times \)16 matrix given below.

\[
\begin{pmatrix}
Y_0^r \\
Y_1^r \\
Y_2^r \\
Y_3^r \\
Y_4^r \\
Y_5^r \\
Y_6^r \\
Y_7^r \\
Y_8^r \\
Y_9^r \\
\vdots \\
Y_{15}^r \\
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
Z_0^r \\
Z_1^r \\
Z_2^r \\
Z_3^r \\
Z_4^r \\
Z_5^r \\
Z_6^r \\
Z_7^r \\
Z_8^r \\
Z_9^r \\
\vdots \\
Z_{15}^r \\
\end{pmatrix}
\]
Previous Cryptanalysis

There are three variants of the block cipher ARIA. The security of the first variant (version 0.8) \[145\] was extensively analyzed by a group of researchers from COSIC, Katholieke Universiteit Leuven \[29\]. In particular, they managed to attack 7-round ARIA using a truncated differential attack. They also had success in applying linear cryptanalysis to 7-round 128-bit key ARIA, and both 10-round 192- and 256-bit key ARIA.

As a result of the attacks, the designers of ARIA proposed a second variant (version 0.9) \[116\]. This made use of four different S-boxes. This variant, reduced to six rounds, was subjected to impossible differentials \[124,178\] and boomerang \[79\] attacks.

The third variant of ARIA (version 1.0) differs from the second variant in the number of rounds and in the key scheduling algorithm. Therefore, attacks on the second variant of ARIA apply also to the third variant if no specific property of the key scheduling algorithm is exploited. Table 2.3 outlines the existing attacks on ARIA and their complexities.

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Attack</th>
<th>Number of keys</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Impossible differential</td>
<td>178</td>
<td>(2^{121}) CP (2^{121}) n/a</td>
</tr>
<tr>
<td>6</td>
<td>Impossible differential</td>
<td>124</td>
<td>(2^{120.5}) CP (2^{104.5}) n/a</td>
</tr>
<tr>
<td>6</td>
<td>Impossible differential</td>
<td>124</td>
<td>(2^{113}) CP (2^{121.6}) n/a</td>
</tr>
<tr>
<td>6</td>
<td>Boomerang</td>
<td>79</td>
<td>(2^{77}) CP (2^{171.2}) n/a</td>
</tr>
<tr>
<td>7</td>
<td>Truncated differential</td>
<td>29</td>
<td>(2^{81}) CP (2^{81}) (2^{64})</td>
</tr>
<tr>
<td>7</td>
<td>Truncated differential</td>
<td>29</td>
<td>(2^{100}) CP (2^{100}) (2^{51})</td>
</tr>
<tr>
<td>7</td>
<td>Linear [29]</td>
<td>29</td>
<td>(2^{77}) KP (2^{88}) (2^{64})</td>
</tr>
<tr>
<td>10</td>
<td>Linear [29]</td>
<td>29</td>
<td>(2^{119}) KP (2^{88}) (2^{64})</td>
</tr>
<tr>
<td>10</td>
<td>Linear [29]</td>
<td>29</td>
<td>(2^{119}) KP (2^{88}) (2^{64})</td>
</tr>
</tbody>
</table>

\[a\] Attack valid for \(2^{96}\) weak keys.
\[b\] Attack valid for \(2^{136}\) weak keys.
\[c\] Attack valid for \(2^{200}\) weak keys.

2.4.3 Noekeon

Noekeon \[60\] is a block cipher submitted to the European NESSIE project \[147\] in 2000. A total of 26 symmetric and 13 asymmetric ciphers were submitted to this project \[148\]. Out of the 26 symmetric ciphers, seven were under the category
of 128-bit block ciphers (which includes Noekeon). Noekeon was not selected to Phase II of NESSIE due to the existence of many related keys discovered by Knudsen and Raddum [111].

Noekeon accepts a 128-bit plaintext block $P$, and a 128-bit master key $K$. The cipher can be implemented in two modes: direct and indirect. In direct mode, the master key is used as the subkey in every round of the cipher. In indirect mode, the master key is used as input to the key scheduling algorithm to produce a set of 17 identical 128-bit round subkeys. The ciphertext block $C$ is produced after $R = 16$ rounds of applying the round function.

Let $P = X^0 = (X_0^0, X_1^0, X_2^0, X_3^0)$ denote the 128-bit plaintext block formed from the concatenation of four 32-bit words $X_r^r$. Let $\hat{K}$ denote the 32-bit round subkey derived from the 128-bit master key $K$. In direct mode, the subkey is the same as the master key ($\hat{K} = K$). Let $L = L_1 \circ L_0$ denote the linear transformation composed of two different linear transformations $L_0$ and $L_1$. The nonlinear transformation is denoted by $S$. These transformations are described in detail later. The encryption algorithm of Noekeon can be described as follows:

$$X^{r+1} = L_1^{-1}(S(L_0(L_0(X^r, \hat{K})))), r = 0, 1, \ldots, 15$$
$$X^{17} = L_0(X^{16}, \hat{K}).$$

Figure 2.7 illustrates the round function of Noekeon. Let $X \ll i$ and $X \gg i$ denote the rotation of the word $X$ by $i$ bits to the left and right, respectively. Let $Y^r = (Y_0^r, Y_1^r, Y_2^r, Y_3^r)$ denote the output block of $L_0$ formed from the concatenation of four 32-bit words $\hat{Y}_i^r$. The linear transformation $L_0$ of Noekeon is described as:

$$\hat{Y}_0^r = X_0^r \oplus RC^r \oplus K_0 \oplus V_0^r \quad (2.6)$$
$$\hat{Y}_1^r = X_1^r \oplus K_1 \oplus V_1^r \quad (2.7)$$
$$\hat{Y}_2^r = X_2^r \oplus K_2 \oplus V_0^r \quad (2.8)$$
$$\hat{Y}_3^r = X_3^r \oplus K_3 \oplus V_1^r \quad (2.9)$$

where $V_0^r = \hat{L}_0(X_1^r \oplus K_1 \oplus X_3^r \oplus K_3)$, $V_1^r = \hat{L}_0(X_0^r \oplus RC^r \oplus X_2^r)$ and $RC^r$ is round $r$ constant.

Let $Y^r = (Y_0^r, Y_1^r, Y_2^r, Y_3^r)$ denote the output block of $L_1$ composed of four 32-bit words $Y_i^r$. The transformation $L_1$ consists simply of three rotations of the
2.4. Analyzed Block Ciphers

Figure 2.7: Round function of Noekeon in Round $r$

words in the input block $\hat{Y}^r$, described as follows.

$$Y^r = L_1(\hat{Y}^r) = (Y^r_0, Y^r_1, Y^r_2, Y^r_3) = (\hat{Y}^r_0, \hat{Y}^r_1 \ll 1, \hat{Y}^r_2 \ll 5, \hat{Y}^r_3 \ll 2).$$

The application of the nonlinear transformation $S$ to $Y^r$ consists of the application of a single $4 \times 4$ S-box $s$ to $Y^r$ 32 times in parallel to produce the output block $Z^r$. Let $Y^r_{i,j}$ denote bit $j$ of the 32-bit word $Y^r_i$. The input word to $s$ consists of the composition of four bits where each bit is taken from each 32-bit word $Y^r_i$ at position $j$ ($0 \leq i \leq 3$). The transformation is given as follows.

$$(Z^r_{0,j}, Z^r_{1,j}, Z^r_{2,j}, Z^r_{3,j}) = s(Y^r_{0,j}, Y^r_{1,j}, Y^r_{2,j}, Y^r_{3,j}), j = 0, 1, \ldots, 31.$$

The inverse of $L_1$, denoted $L_1^{-1}$, uses the block $Z^r$ as input to produce the output block $X^{r+1}$, given as follows.

$$X^{r+1} = L_1^{-1}(Z^r) = (X^{r+1}_0, X^{r+1}_1, X^{r+1}_2, X^{r+1}_3) = (Z^r_0 \gg 1, Z^r_1 \gg 5, Z^r_2 \gg 5, Z^r_3 \gg 2).$$

**Previous Cryptanalysis**

Currently, the only cryptanalysis of Noekeon in the public literature is the existence of related-key differentials on the full cipher, revealed by Knudsen and Raddum [111]. Specifically, if a key $K$ encrypts the plaintext $P$ to the ciphertext
Chapter 2. Symmetric Ciphers

Then there exist another key $K' = K \oplus \Delta_K$ that encrypts $P' = P \oplus \Delta_D$ to $C' = C \oplus \Delta_D$ with probability $2^{-32}$. However, they note that it is not clear how to exploit these properties in an attack.

2.4.4 Serpent

Serpent is one of the block ciphers submitted to the AES process in 1998 [4]. In August 1999, Serpent was selected as one of the five finalists [143]. Despite failing to become the AES, the cipher is still being actively analyzed by the cryptographic community since 2000 [14 17 43 49 55 65 76 85 99 113].

Serpent accepts a 128-bit plaintext block $P$ and a master key $K$ with sizes of 128, 192 or 256 bits. If the size of the master key is less than 256 bits, then the key is first padded with a one ‘1’ bit and followed by as many ‘0’ bits as necessary to form a 256-bit key. The key scheduling algorithm uses the master key (or the padded master key) as input to generate a set of thirty-two 128-bit round subkeys $K_r$ ($0 \leq r \leq 31$). The ciphertext block $C$ is constructed after applying the round function 32 times. The cipher can be represented in non-bit-sliced and bit-sliced version. This thesis focuses on the bit-sliced version of Serpent.

The round function is composed of a key mixing transformation, a nonlinear transformation $S_r$, and a linear transformation $L$. In the last round, the transformation $L$ is replaced by a key mixing transformation. The encryption algorithm of Serpent can be expressed by the following equations:

\[
X^{r+1} = L(S_{r \mod 8}(X^r \oplus K^r)), \quad r = 0, 1, \ldots, 30
\]
\[
X^{32} = S_7(X^{31} \oplus K^{31}) \oplus K^{32}.
\]

The round function of Serpent in Round $r$ is depicted in Figure 2.8.

Serpent uses eight different nonlinear transformations $S_{r \mod 8}$ ($0 \leq r \leq 7$). Let $Y^r$ denote the output block of the key mixing transformation, i.e. $Y^r = X^r \oplus K^r$. The nonlinear transformation $S_r$ applies a single 4 × 4 S-box $s_r$ to $Y^r$ 32 times in parallel to produce the output block $Z^r$. The input word to $s_r$ is the concatenation of four bits where a bit is taken from position $j$ of each 32-bit word $Y^r_i$. This transformation is given as follows.

\[
(Z_{0,j}^r, Z_{1,j}^r, Z_{2,j}^r, Z_{3,j}^r) = s_{r \mod 32}(Y_{0,j}^r, Y_{1,j}^r, Y_{2,j}^r, Y_{3,j}^r), \quad j = 0, 1, \ldots, 31.
\]
Let $X^{r+1} = (X_0^{r+1}, X_1^{r+1}, X_2^{r+1}, X_3^{r+1})$ denote the output block of $L$ composed of four 32-bit words $X_i^{r+1}$. The transformation $L$ of Serpent uses the block $Z^r$ as input to produce the output block $X^{r+1}$, i.e. $X^{r+1} = L(Z^r)$. Specifically, this is expressed as follows for $j = 0, 1, \ldots, 31$:

\[
\begin{align*}
X_0^{r+1} &= Z_0^r \oplus Z_1^r \oplus Z_2^r \oplus Z_3^r \\
X_1^{r+1} &= Z_1^r \oplus Z_0^r \oplus Z_2^r \\
X_2^{r+1} &= Z_2^r \oplus Z_3^r \oplus Z_0^r \oplus Z_1^r \\
X_3^{r+1} &= Z_3^r \oplus Z_2^r \oplus Z_0^r \\
\end{align*}
\]

where the indices $j - k$ are computed modulo 32 and $j \ll k$ denotes the shifting of $j$ by $k$ bits to the left. This means that if $(j \ll k) = j - k$ is less than zero, then the bit at position $j$ is a zero bit. Similarly, the inverse of $L$, denoted $L^{-1}$,
Table 2.4: The linear transformation of Serpent. The output bits $\tilde{X}^{r+1}$ are displayed as the XOR sum of the input bits $\tilde{Z}_j^r$. For instance, $\tilde{X}^{r+1} = \tilde{Z}_0^r \oplus \tilde{Z}_8^r \oplus \tilde{Z}_{127}^r$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$X^{r+1}_{i,32}$</th>
<th>$X^{r+1}_{i,64}$</th>
<th>$X^{r+1}_{i,96}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18, 63, 82</td>
<td>19, 21, 34, 64, 71, 92, 99</td>
<td>9, 88, 121</td>
</tr>
<tr>
<td>1</td>
<td>19, 32, 83</td>
<td>20, 22, 35, 65, 72, 93, 100</td>
<td>10, 87, 122</td>
</tr>
<tr>
<td>2</td>
<td>20, 33, 84</td>
<td>21, 23, 36, 66, 73, 94, 101</td>
<td>11, 88, 123</td>
</tr>
<tr>
<td>3</td>
<td>21, 34, 85</td>
<td>22, 24, 37, 67, 74, 95, 102</td>
<td>12, 89, 124</td>
</tr>
<tr>
<td>4</td>
<td>22, 35, 86</td>
<td>23, 25, 38, 68, 64, 75, 103</td>
<td>13, 90, 125</td>
</tr>
<tr>
<td>5</td>
<td>23, 36, 87</td>
<td>24, 26, 39, 69, 65, 76, 104</td>
<td>14, 91, 126</td>
</tr>
<tr>
<td>6</td>
<td>24, 37, 88</td>
<td>25, 27, 40, 66, 70, 77, 105</td>
<td>15, 92, 127</td>
</tr>
<tr>
<td>7</td>
<td>25, 38, 89</td>
<td>26, 28, 41, 67, 71, 78, 106</td>
<td>93, 96</td>
</tr>
<tr>
<td>8</td>
<td>26, 39, 90</td>
<td>27, 29, 42, 68, 72, 79, 107</td>
<td>94, 97</td>
</tr>
<tr>
<td>9</td>
<td>27, 40, 91</td>
<td>28, 30, 43, 69, 73, 80, 108</td>
<td>95, 98</td>
</tr>
<tr>
<td>10</td>
<td>28, 41, 92</td>
<td>29, 31, 44, 70, 74, 81, 109</td>
<td>96, 99</td>
</tr>
<tr>
<td>11</td>
<td>29, 42, 93</td>
<td>30, 31, 45, 71, 75, 82, 110</td>
<td>20, 65, 100</td>
</tr>
<tr>
<td>12</td>
<td>30, 43, 94</td>
<td>31, 32, 46, 72, 76, 83, 111</td>
<td>21, 66, 101</td>
</tr>
<tr>
<td>13</td>
<td>31, 44, 95</td>
<td>32, 33, 47, 73, 77, 84, 112</td>
<td>22, 67, 102</td>
</tr>
<tr>
<td>14</td>
<td>32, 45, 96</td>
<td>33, 34, 48, 74, 78, 85, 113</td>
<td>23, 68, 103</td>
</tr>
<tr>
<td>15</td>
<td>33, 46, 97</td>
<td>34, 35, 49, 75, 79, 86, 114</td>
<td>24, 69, 104</td>
</tr>
<tr>
<td>16</td>
<td>34, 47, 98</td>
<td>35, 36, 50, 76, 80, 87, 115</td>
<td>25, 70, 105</td>
</tr>
<tr>
<td>17</td>
<td>35, 48, 99</td>
<td>36, 37, 51, 77, 81, 88, 116</td>
<td>26, 71, 106</td>
</tr>
<tr>
<td>18</td>
<td>36, 49, 100</td>
<td>37, 38, 52, 78, 82, 89, 117</td>
<td>27, 72, 107</td>
</tr>
<tr>
<td>19</td>
<td>37, 50, 101</td>
<td>38, 39, 53, 79, 83, 90, 118</td>
<td>28, 73, 108</td>
</tr>
<tr>
<td>20</td>
<td>38, 51, 102</td>
<td>39, 40, 54, 80, 84, 91, 119</td>
<td>29, 74, 109</td>
</tr>
<tr>
<td>21</td>
<td>39, 52, 103</td>
<td>40, 41, 55, 81, 85, 92, 120</td>
<td>30, 75, 110</td>
</tr>
<tr>
<td>22</td>
<td>40, 53, 104</td>
<td>41, 42, 56, 82, 86, 93, 121</td>
<td>31, 76, 111</td>
</tr>
<tr>
<td>23</td>
<td>41, 54, 105</td>
<td>42, 43, 57, 83, 87, 94, 122</td>
<td>32, 77, 112</td>
</tr>
<tr>
<td>24</td>
<td>42, 55, 106</td>
<td>43, 44, 58, 84, 88, 95, 123</td>
<td>33, 78, 113</td>
</tr>
<tr>
<td>25</td>
<td>43, 56, 107</td>
<td>44, 45, 59, 85, 89, 96, 124</td>
<td>34, 79, 114</td>
</tr>
<tr>
<td>26</td>
<td>44, 57, 108</td>
<td>45, 46, 60, 86, 90, 97, 125</td>
<td>35, 80, 115</td>
</tr>
<tr>
<td>27</td>
<td>45, 58, 109</td>
<td>46, 47, 61, 87, 91, 98, 126</td>
<td>36, 81, 116</td>
</tr>
<tr>
<td>28</td>
<td>46, 59, 110</td>
<td>47, 48, 62, 88, 92, 99, 127</td>
<td>37, 82, 117</td>
</tr>
<tr>
<td>29</td>
<td>47, 60, 111</td>
<td>48, 49, 63, 89, 93, 100, 128</td>
<td>38, 83, 118</td>
</tr>
<tr>
<td>30</td>
<td>48, 61, 112</td>
<td>49, 50, 64, 90, 94, 101, 129</td>
<td>39, 84, 119</td>
</tr>
<tr>
<td>31</td>
<td>49, 62, 113</td>
<td>50, 51, 65, 91, 95, 102, 130</td>
<td>40, 85, 120</td>
</tr>
</tbody>
</table>

is described as follows:

$$Z^{r+1}_{0,j} = X^{r+1}_{0,j,18} \oplus X^{r+1}_{1,j,13} \oplus X^{r+1}_{3,j,13}$$

$$Z^{r+1}_{1,j} = X^{r+1}_{1,j,1} \oplus X^{r+1}_{0,j,5} \oplus X^{r+1}_{1,j} \oplus X^{r+1}_{2,j,22} \oplus X^{r+1}_{1,j} \oplus X^{r+1}_{3,j}$$

$$Z^{r+1}_{2,j} = X^{r+1}_{2,j,25} \oplus X^{r+1}_{3,j,3} \oplus X^{r+1}_{1,j,3}$$

$$Z^{r+1}_{3,j} = X^{r+1}_{3,j,7} \oplus X^{r+1}_{2,j,22} \oplus X^{r+1}_{3,j} \oplus X^{r+1}_{1,j,7} \oplus X^{r+1}_{0,j,3} \oplus X^{r+1}_{1,j,3} \oplus X^{r+1}_{3,j,3}$$

If the input word $Z^r = (Z^r_0, Z^r_1, \ldots, Z^r_{127})$ and output word $X^{r+1} = (X^{r+1}_0, X^{r+1}_1, \ldots, X^{r+1}_{127})$ are depicted as the concatenation of 128 bits $Z^r_i$ and $X^{r+1}_i$, respectively, then $L$ can be represented as a nonsingular $128 \times 128$ binary matrix. Table 2.4 depicts the output bits $\tilde{X}^{r+1}$ of the transformation $L$ of Serpent as the XOR sum of particular input bits $\tilde{Z}_j^r$. 
Previous Cryptanalysis

The cipher has endured extensive cryptanalysis including differential, linear, differential-linear, boomerang and rectangle attacks. Several existing attacks on Serpent are given in Table 2.5. To date, there are attacks on six to 11 rounds of Serpent [14, 17, 48, 99, 113]. The best existing attack is a differential-linear attack on Serpent reduced to 12 rounds [67]. The attack requires $2^{123.5}$ chosen plaintexts and $2^{249.4}$ encryptions and thus, valid for only the 256-bit key variant of Serpent.

Table 2.5: Summary of existing attacks on Serpent

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Attack</th>
<th>Complexity</th>
<th>Data</th>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Integral (Chapter 3)</td>
<td>$2^{11}$ CP</td>
<td>$2^{20}$</td>
<td>small</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Integral (Chapter 3)</td>
<td>$2^{13.2}$ CP</td>
<td>$2^{65.2}$</td>
<td>$2^{37}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Differential [113]</td>
<td>$2^{283}$ CP</td>
<td>$2^{190}$</td>
<td>$2^{44}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Differential [113]</td>
<td>$2^{271}$ CP</td>
<td>$2^{103}$</td>
<td>$2^{79}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Differential [113]</td>
<td>$2^{241}$ CP</td>
<td>$2^{163}$</td>
<td>$2^{49}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Integral (Chapter 3)</td>
<td>$2^{65.2}$ CP</td>
<td>$2^{112.3}$</td>
<td>$2^{37}$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Differential [14]</td>
<td>$2^{284}$ CP</td>
<td>$2^{206.7}$</td>
<td>$2^{89}$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Amplified boomerang [99]</td>
<td>$2^{110}$ CP</td>
<td>$2^{252}$</td>
<td>$2^{212}$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Rectangle [16]</td>
<td>$2^{126.3}$ CP</td>
<td>$2^{165}$</td>
<td>$2^{131.8}$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Boomerang [16]</td>
<td>$2^{126.3}$ ACPC</td>
<td>$2^{165}$</td>
<td>$2^{89}$</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Linear [15]</td>
<td>$2^{118}$ KP</td>
<td>$2^{205.7}$</td>
<td>$2^{183}$</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Differential-linear [17]</td>
<td>$2^{125.3}$ CP</td>
<td>$2^{172.4}$</td>
<td>$2^{30}$</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Differential-linear [17]</td>
<td>$2^{125.3}$ CP</td>
<td>$2^{139.2}$</td>
<td>$2^{60}$</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Linear [48]</td>
<td>$2^{118}$ KP</td>
<td>$2^{178}$</td>
<td>$2^{88}$</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Differential-linear [67]</td>
<td>$2^{123.5}$ CP</td>
<td>$2^{249.4}$</td>
<td>$2^{128.5}$</td>
<td></td>
</tr>
</tbody>
</table>

2.4.5 PRESENT

PRESENT is a block cipher proposed in 2007 targeted for application in hardware-constraint environments such as RFID tags and sensor networks [37]. The cipher has a very simple design and has attracted a lot of attention from the cryptographic community [3, 17, 39, 150, 152, 174].

PRESENT accepts a 64-bit plaintext block $P$ and master key $K$ of sizes 80 and 128 bits. The master key is used by the key scheduling algorithm as input to produce a set of thirty-two 64-bit round subkeys $K^r \ (0 \leq r \leq 31)$. The ciphertext block $C$ is generated after applying the round function 31 times and
an additional key mixing transformation.

The round function of PRESENT consists of a key mixing transformation, a nonlinear transformation $S$ and a linear transformation $L$. These transformations are described in detail later. Figure 2.9 illustrates the round function of this cipher. The encryption algorithm of PRESENT is given as:

$$X^{r+1} = L(S(X^r \oplus K^r)), \ r = 0, 1, \ldots, 30$$
$$X^{32} = X^{31} \oplus K^{31}$$

Let $Y^r$ denote the output block of the key mixing transformation in round $r$. The nonlinear transformation $S$ applies a single $4 \times 4$ S-box $s$ to $Y^r$ 16 times in parallel to produce the output block $Z^r$. This transformation is described as follows

$$Z^r = S(Y^r) = (Z^r_0, Z^r_1, Z^r_2, Z^r_3) = (s(Y^r_0), s(Y^r_1), s(Y^r_2), s(Y^r_3)).$$

Let $X^{r+1} = (X_0^{r+1}, X_1^{r+1}, X_2^{r+1}, X_3^{r+1})$ denote the output block of $L$ composed of four 16-bit words $X_i^{r+1}$. The block $Z^r$ is used as input to $L$ to produce the output block $X^{r+1}$, i.e. $X^{r+1} = L(Z^r)$. The transformation $L$ of PRESENT is a
2.4. Analyzed Block Ciphers

simple bit permutation given by the following equations.

\[
X^{r+1}_0 = (Z^r_{0,0}, Z^r_{0,4}, Z^r_{0,8}, Z^r_{1,12}, Z^r_{1,14}, Z^r_{1,18}, Z^r_{1,12}, Z^r_{2,0}, Z^r_{2,4}, Z^r_{2,8}, Z^r_{2,12}, Z^r_{3,0}, Z^r_{3,4}, Z^r_{3,8}, Z^r_{3,12})
\]

\[
X^{r+1}_1 = (Z^r_{0,1}, Z^r_{0,5}, Z^r_{0,9}, Z^r_{1,13}, Z^r_{1,11}, Z^r_{1,5}, Z^r_{1,9}, Z^r_{1,13})
\]

\[
X^{r+1}_2 = (Z^r_{2,1}, Z^r_{2,5}, Z^r_{2,9}, Z^r_{2,13}, Z^r_{3,1}, Z^r_{3,5}, Z^r_{3,9}, Z^r_{3,13})
\]

\[
X^{r+1}_3 = (Z^r_{0,2}, Z^r_{0,6}, Z^r_{0,10}, Z^r_{0,14}, Z^r_{1,2}, Z^r_{1,6}, Z^r_{1,10}, Z^r_{1,14})
\]

\[
X^{r+1}_4 = (Z^r_{2,2}, Z^r_{2,6}, Z^r_{2,10}, Z^r_{2,14}, Z^r_{3,2}, Z^r_{3,6}, Z^r_{3,10}, Z^r_{3,14})
\]

\[
X^{r+1}_5 = (Z^r_{0,3}, Z^r_{0,7}, Z^r_{0,11}, Z^r_{0,15}, Z^r_{1,3}, Z^r_{1,7}, Z^r_{1,11}, Z^r_{1,15})
\]

\[
X^{r+1}_6 = (Z^r_{2,3}, Z^r_{2,7}, Z^r_{2,11}, Z^r_{2,15}, Z^r_{3,3}, Z^r_{3,7}, Z^r_{3,11}, Z^r_{3,15})
\]

Previous Cryptanalysis

Several existing attacks on Present are given in Table 2.6. The cipher has been attacked using differential [174], related-key rectangle [152] and algebraic-differential [3] attacks. Currently, the best existing attack on Present is a statistical saturation attack on 24 rounds [47].

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Attack</th>
<th>Number of keys</th>
<th>Complexity of keys</th>
<th>Data</th>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Integral (Chapter 3)</td>
<td>1</td>
<td>2^{24.3} CP</td>
<td>2^{100.1}</td>
<td>2^{77}</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Related-key rectangle [152]</td>
<td>4</td>
<td>2^{63} RK-CP</td>
<td>2^{104}</td>
<td>2^{53}</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Algebraic-differential [3]</td>
<td>1</td>
<td>6 \cdot 2^{62} CP</td>
<td>2^{113}</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Statistical saturation [47]</td>
<td>1</td>
<td>2^{60} CP</td>
<td>2^{20}</td>
<td>2^{16}</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Statistical saturation [47]</td>
<td>1</td>
<td>2^{57} CP</td>
<td>2^{57}</td>
<td>2^{32}</td>
<td></td>
</tr>
</tbody>
</table>

2.4.6 SMS4

SMS4 [64,149] is a standard block cipher used in the Chinese Wireless Local Area Network (WLAN) Wired Authentication and Privacy Infrastructure (WAPI). The WAPI was proposed to the International Organization for Standardization (ISO) as an international WLAN standard. However, in 2006, the ISO adopted IEEE 802.11i as the standard and WAPI was rejected [126]. The rejection is partially due to the uncertainty regarding the security of SMS4 because during
that time, the specification was undisclosed. Despite being rejected by the ISO, the WAPI is still widely used in China. In January 2006, the specification of the cipher was made public and has since been subjected to extensive cryptanalysis [74, 104, 125, 126, 168, 183].

SMS4 accepts a 128-bit plaintext block $P$, and a 128-bit master key $K$. The master key is used as input to the key scheduling algorithm to produce a set of thirty-two 32-bit round subkeys. The plaintext block and the round subkeys are used as input to the encryption algorithm to produce the ciphertext block $C$. The encryption algorithm consists of 32 applications of the round function. Using the terminology of Schneier and Kelsey, the cipher employs a homogeneous, complete, source-heavy unbalanced Feistel network structure [159]. For simplicity, the notation used to depict the state words of SMS4 is different from the notation used in previous block ciphers.

**Encryption Algorithm**

Let $P = (X_0, X_1, X_2, X_3)$ denote the 128-bit plaintext block formed from the concatenation of four 32-bit words $X_i$. Let $K_i$ denote the 32-bit $i$-th round subkey derived from the 128-bit master key $K$. The derivation of these subkeys is explained in Section 2.4.6. Let $T = L \circ S$ denote the function composed of the nonlinear transformation $S$ and the linear transformation $L$. Both $S$ and $L$ are described in detail later. The $i$-th round function of the encryption algorithm can be described as follows:

$$X_{i+4} = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus K_i), \ i = 0, 1, \ldots, 31$$

and is depicted in Figure 2.10. The ciphertext consists of the concatenation of the four 32-bit words $C = (X_{35}, X_{34}, X_{33}, X_{32})$. It is obtained in the reverse order from the output of the final round function to facilitate decryption.

Decryption is the same as encryption with the only difference being the order in which the subkeys are used; this is in the reverse order as follows:

$$X_i = X_{i+4} \oplus T(X_{i+3} \oplus X_{i+2} \oplus X_{i+1} \oplus K_i), \ i = 31, 30, \ldots, 0.$$
2.4. Analyzed Block Ciphers

The application of the nonlinear transformation $S$ to $X_i$ consists of the application of a single $8 \times 8$ S-box $s$ to $X_{i,j}$ as follows:

$$S(X_i) = (s(X_{i,0}), s(X_{i,1}), s(X_{i,2}), s(X_{i,3})).$$

The linear transformation $L$ is defined as:

$$L(X_i) = X_i \oplus (X_i \ll 2) \oplus (X_i \ll 10) \oplus (X_i \ll 18) \oplus (X_i \ll 24).$$

**Key Scheduling Algorithm**

In the initialization phase of the key scheduling algorithm, a 128-bit constant $FK$ is XORed with the 128-bit master key $K$ to produce the initial inputs for the key scheduling algorithm. Let $K = (MK_0, MK_1, MK_2, MK_3)$ denote the master key formed from the concatenation of four 32-bit words $MK_i$. Similarly, let $FK = (FK_0, FK_1, FK_2, FK_3)$ denote the constant as the concatenation of four 32-bit words $FK_i$, where $FK_0 = \text{A3B1BAC6}$, $FK_1 = \text{56AA3350}$, $FK_2 = \text{677D9197}$ and $FK_3 = \text{B27022DC}$ (in hexadecimal). Then, the initial input words to the key scheduling algorithm are $K_{i-4} = MK_i \oplus FK_i$ for $i = 0, 1, 2, 3$. Note that this initialization phase has no cryptographic significance because the operation is linear and the constants are known.

Let $T' = L' \circ S$ denote the function composed of the nonlinear transformation $S$ and the linear transformation $L'$ ($L'$ is described later). Note that this transformation $L'$ is the only difference between the structure of the encryption and the key scheduling algorithms. Let $K_i$ denote the 32-bit subkey and $CK_i$ denote the 32-bit constant in round $i$. The $i$-th round function of the key scheduling algorithm can be described as follows:

$$K_i = K_{i-4} \oplus T'(K_{i-3} \oplus K_{i-2} \oplus K_{i-1} \oplus CK_i), \ i = 0, 1, \ldots, 31.$$
The round constants $C_k = (C_{k,0}, C_{k,1}, C_{k,2}, C_{k,3})$, which are composed of the concatenation of four 8-bit words $C_{k,j}$, are defined as

$$C_{k,j} = (28i + 7j) \mod 256, \ i = 0, 1, \ldots, 31 \text{ and } j = 0, 1, 2, 3.$$ 

The linear transformation $L'$ is defined as:

$$L'(X) = X \oplus (X \ll 13) \oplus (X \ll 23).$$

### Table 2.7: Summary of existing attacks on the SMS4 block cipher

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Attack</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Integral [125]</td>
<td>$2^{16}$ CP</td>
</tr>
<tr>
<td>14</td>
<td>Rectangle [126]</td>
<td>$2^{121.82}$ CP</td>
</tr>
<tr>
<td>14</td>
<td>Rectangle 168</td>
<td>$2^{107.89}$ CP</td>
</tr>
<tr>
<td>16</td>
<td>Rectangle 183</td>
<td>$2^{11} CP$</td>
</tr>
<tr>
<td>16</td>
<td>Impossible differential [126]</td>
<td>$2^{11} CP$</td>
</tr>
<tr>
<td>16</td>
<td>Impossible differential [168]</td>
<td>$2^{117.06}$ CP</td>
</tr>
<tr>
<td>18</td>
<td>Boomerang [104]</td>
<td>$2^{120}$ ACPC</td>
</tr>
<tr>
<td>18</td>
<td>Rectangle 104</td>
<td>$2^{124}$ CP</td>
</tr>
<tr>
<td>21</td>
<td>Differential 104</td>
<td>$2^{118}$ CP</td>
</tr>
<tr>
<td>22</td>
<td>Differential 104</td>
<td>$2^{118}$ CP</td>
</tr>
<tr>
<td>22</td>
<td>Linear 104</td>
<td>$2^{117}$ KP</td>
</tr>
<tr>
<td>22</td>
<td>Linear 74</td>
<td>$2^{119}$ KP</td>
</tr>
</tbody>
</table>

Previous Cryptanalysis

All previous attacks on the SMS4 block cipher fall into the category of single-key attacks. The cipher has been attacked using integral, rectangle, impossible differential, boomerang, differential and linear cryptanalysis [74,104,125,126,168,183]. The best existing attacks on SMS4 are differential and linear cryptanalysis on 22 rounds [74,104,183]. Table 2.7 outlines the previous attacks on SMS4 with their complexities.

### 2.4.7 LEX-AES

LEX is a generic method proposed for constructing a stream cipher from a block cipher initially introduced by Biryukov in 2005 [26]. The method was submitted
to eSTREAM, the ECRYPT Stream Cipher Project. The basic idea of LEX is to use a block cipher as a keystream generator for a binary additive stream cipher. The keystream is produced by extracting part of the internal state at specific rounds. In the LEX proposal, an example of using this method is given where the AES is selected as the block cipher. In the remainder of this thesis, the term LEX-AES is used to refer to this specific instance based on the AES with 128-bit key. Refer to Section 2.4.1 for the description of the AES.

There are two versions of LEX-AES. The first version uses the full AES in both the IV setup and the state update function [20]. This similarity was exploited in a slide attack [177]. In order to resist this attack, a second version of LEX-AES was proposed in 2007 [27]. This version uses the full AES in the IV setup but a slightly modified AES in the state update function. This version was vulnerable to a key recovery attack [68, 69] which resulted in the cipher being dropped from the final eSTREAM portfolio [6]. The version of LEX-AES examined in this chapter is based on this second version.

LEX-AES is analyzed in this thesis because the application of the LEX method gives a glimpse of values that would be the intermediate calculation of a block cipher. This is interesting since these values are not available to a cryptanalyst in normal block cipher modes.

LEX-AES has an internal state of 32 bytes, composed of a 16-byte state block $X$ and a 16-byte secret key block $K$. The functions which LEX-AES uses to initialize the internal state and the process by which the keystream is generated are described later in this section, and illustrated in Figure 2.11.
Initialization

The 128-bit secret key \( K \) is expanded by the key scheduling algorithm to produce eleven 128-bit round subkeys, denoted \( K^0, K^1, \ldots, K^{10} \). All 11 round subkeys are used in the initialization process but only 10 round subkeys are used by the state update function during keystream generation.

The internal state is initialized by encrypting a 128-bit IV with \( K \) using the full 10-round AES. According to Biryukov [27], the encrypted IV is concatenated with \( K \) to form the initial 256-bit internal state of LEX-AES, i.e. \( (X, K) \). However, as explained later in this section, only \( X \) is updated during keystream generation. Let \( X^t \) denote the value of the block \( X \) at iteration \( t \) and \( E_K \) denote the AES encryption using the key \( K \). The initialization process is given as follows:

\[
X^1 = E_K(IV) \oplus K^0
\]

Keystream Generation

The 16-byte internal state \( X^t = (X^t_0, X^t_1, \ldots, X^t_{15}) \) is depicted as a \( 4 \times 4 \) matrix. The content of this state is updated using the round function of the AES denoted by \( F_{K^t} \). The \( T \) iterations of the keystream generator of LEX-AES can be described by the following algorithm:

\[
X^{t+1} = F_{K^{t \mod 10}}(X^t) \quad t = 1, 2, \ldots, T
\]

In each iteration, after the state is updated, four bytes of \( X^t \) are leaked directly by the output function \( f \) to form the keystream. The positions of the leaked bytes are dependent on whether the round number is odd or even. Figure 2.12 shows the different leak positions, which are shaded in gray. The output function \( f \) performs the following operations when producing the keystream:

\[
f(X^t) = \begin{cases} 
(X^t_0, X^t_2, X^t_8, X^t_{10}) & \text{if } t \text{ is odd} \\
(X^t_4, X^t_6, X^t_{12}, X^t_{14}) & \text{if } t \text{ is even}
\end{cases}
\]

In LEX-AES, the state update function is the AES round function, which transforms \( X^t \) into \( X^{t+1} \). The round function is key-dependent, and makes use of 10 round subkeys \( K^0, \ldots, K^9 \). After 10 iterations, the state update function reuses the 10 subkeys, provided the secret key has remained unchanged. Every 10th 4-byte output is therefore produced using the same subkey. In a more secure
2.5 Summary and Conclusion

This chapter presents review of the literature that is relevant to this thesis. It is explained that the basic building blocks of a block cipher include the linear and nonlinear transformations. These transformations respectively provide the crucial diffusion and confusion properties described by Shannon [162]. The interaction between these two main transformations are exploited in some way by the attacks presented in this chapter.

In terms of real-world applications, linear and algebraic cryptanalysis are more practical than differential-type cryptanalysis. This is because linear and algebraic cryptanalysis are known plaintext attacks whereas differential-type cryptanalysis fall into the category of chosen plaintext attacks. The least practical attacks are the related-key-type cryptanalysis. Despite this, related-key-type cryptanalysis have been shown, in theory, to be applicable to full rounds version

Odd rounds

<table>
<thead>
<tr>
<th>a0</th>
<th>a4</th>
<th>a8</th>
<th>a12</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>a5</td>
<td>a9</td>
<td>a13</td>
</tr>
<tr>
<td>a2</td>
<td>a6</td>
<td>a10</td>
<td>a14</td>
</tr>
<tr>
<td>a3</td>
<td>a7</td>
<td>a11</td>
<td>a15</td>
</tr>
</tbody>
</table>

Even rounds

<table>
<thead>
<tr>
<th>a0</th>
<th>a4</th>
<th>a8</th>
<th>a12</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>a5</td>
<td>a9</td>
<td>a13</td>
</tr>
<tr>
<td>a2</td>
<td>a6</td>
<td>a10</td>
<td>a14</td>
</tr>
<tr>
<td>a3</td>
<td>a7</td>
<td>a11</td>
<td>a15</td>
</tr>
</tbody>
</table>

Figure 2.12: Different leaks in odd and even rounds

variant of LEX-AES, the secret key $K$ is changed at least every $2^{32}$ IV setups and the IV is changed every $T = 500$ iterations. This means that one unique key and IV pair is allowed to produce only $500 \cdot 4 = 2000$ bytes of keystream.

Previous Cryptanalysis

As mentioned earlier in this section, there are two versions of LEX-AES. The first version [26] was susceptible to a slide attack. The attack enabled a particular key to be recovered if used with about $2^{61}$ random IVs where each IV produces 20,000 bytes of keystream [177].

The second version of LEX-AES [27] was subjected to a key recovery attack by Dunkelman and Keller which requires $2^{36.3}$ bytes of keystream and $2^{112}$ operations [68, 69]. They note that their attack can also be applied to the first version of LEX-AES.
of various block ciphers, including, recently, the AES [31, 32]. Note, however, that due to the requirement of the attack, which involves choosing the relation between two or more unknown keys, the attack is not very practical to be implemented in the real world.

Three of the block ciphers examined, which are the AES, ARIA and SMS4, are used as standard respectively in the United States, Korea and China. Noekeon is a block cipher submitted to the NESSIE open process. The block cipher PRESENT represents a breed of ciphers targeted at lightweight applications and LEX-AES is based primarily on the AES block cipher. Since some of these ciphers have already been used in real-world applications and some are new proposals, the security of these ciphers have to remain current. This is one of the reasons why these specific block ciphers are chosen for examination in subsequent chapters.
Chapter 3

Integral Attack on Bit-Based Block Ciphers

The integral attack \cite{112}, which was discussed in Section 2.3.6, is the basis for the best attacks on the Advanced Encryption Standard (AES) \cite{76,128} to date, and has become a standard inclusion in a cryptanalyst’s toolbox. The attack exploits the linear relationship between texts at intermediate stages of encryption. The existence of this linear relationship is used to verify partial subkey bits in a key recovery attack.

The integral attack is intended mainly for application to word-based block ciphers. To date, the attack has not been thought suitable for application to bit-based block ciphers. A word-based block cipher treats the output bits of an S-box as a word. Therefore, the subsequent linear transformation mixes these words in a way that respects the S-box boundaries. A bit-based block cipher treats the output bits of an S-box independently. Therefore, the subsequent linear transformation mixes these bits in a way that may extend beyond the S-box boundaries. This bit-based behaviour impedes the propagation of useful word-based properties given in Section 2.3.6. For instance, the integral property $A$ of any sub-block in a structure is normally preserved by a word-based linear transformation. In contrast, this property $A$ will be subsequently destroyed by a bit-based linear transformation. The useful properties of this structure will then be lost after one round. This problem is illustrated in Example 3.1.
Example 3.1 Consider the round functions of two 16-bit block ciphers, one bit-based and one word-based given in Figures 3.1 and 3.2 respectively. In these figures, the lines denote a one bit data path and $s$ denotes a $4 \times 4$ bijective S-box.

Additionally, consider a structure of sixteen text blocks that has the integral property ACCC, where each text block is partitioned into four 4-bit sub-blocks. If this structure is used as input to the round function of the word-based block cipher, then after the linear transformation, the structure has the integral property CCCA. If, however, the structure is used as input to the round function of the bit-based block cipher, then after the linear transformation, the structure has the integral property BBBB. This clearly shows that the bit-based linear transformation destroys property A of the structure and prevents the application of traditional integral attacks.

In this chapter, the traditional integral attack is extended by proposing new methods to apply the attack to bit-based block ciphers. In this approach, the sub-blocks of the structure are further partitioned into bits. Each bit of the structure holds a specific sequence of bits ‘0’ and ‘1’. The pattern of the bit sequence serves as the basis of the notation. This means that the order of the texts in a bit-pattern based integral attack plays an important role, in contrast to the usual integral attack where the texts are regarded as an unordered set. The refined notation allows an attacker to gain knowledge of bit patterns in the structure through some encryption rounds. Instead of having all possible values as input to a single S-box in the first round, the bit-pattern based structure is constructed such that the active bits are spread over more than one S-box.

This bit-pattern based integral attack is shown to successfully attack up to 5 (out of 16) rounds of Noekeon [60], 6 (out of 32) rounds of Serpent [4] and 7 (out of 31) rounds of Present [37]. All attacks rely on the existence of linear relationships implied by the bit patterns at intermediate stages of encryption. It
3.1 Bit-Pattern-Based Integral Attack

is believed that this is the first integral cryptanalysis on Noekeon and PRESENT reported in the literature.

This chapter is organized as follows. The bit-based integral attack and the new refined notations are explained in Section 3.1. The application of the attacks to Noekeon, Serpent and PRESENT are presented in Section 3.2. Experimental results of these applications are given in Section 3.3. A discussion of these results is presented in Section 3.4. Section 3.5 highlights some related work. A summary of the work in this chapter and conclusions are given in Section 3.6.

3.1 Bit-Pattern-Based Integral Attack

This section introduces the notations used for specific bit patterns and explains how to trace these patterns through the linear and nonlinear transformations. The generic integral attack on bit-based block ciphers is also presented.

3.1.1 The Bit-Pattern-Based Notations

In this bit-pattern based approach, a particular sequence of individual bits of a structure of texts is treated independently. If the texts in the structure are considered as row vectors in a binary matrix, then the bit patterns represent the bit sequence of the column vectors. The pattern of the bit sequence forms the foundation of the bit-based notations described as follows.

- The pattern $c$ means that all bits in this position within the structure consist of either bit ‘0’ or ‘1’. This pattern is called a constant bit pattern.

- The pattern $a_i$ means that the bits in the first block of $2^i$ consecutive bits in this position are all zeros or all ones, and the next block of $2^i$ consecutive bits all have the opposite value of the previous block. The alternating values of bits in $2^i$-blocks is repeated throughout the structure. This pattern is called an active bit pattern.

- The pattern $b_i$ means that the bits in blocks of $2^i$ consecutive bits in this position are all zeros or all ones, but the values of the $2^i$-blocks are not necessarily repeated in an alternating manner.

- The pattern $d_i$ means that the bits in this position may hold either a $c$ (constant) or an $a_i$ (active) pattern. This pattern is called a dual bit
Table 3.1: Examples of text values for $c$, $a_i$ and selected $b_i$ bit patterns in a structure of sixteen texts where $x, y \in \mathbb{F}_2$ and $y = \bar{x}$

<table>
<thead>
<tr>
<th>$c$</th>
<th>$a_3$</th>
<th>$a_2$</th>
<th>$a_1$</th>
<th>$a_0$</th>
<th>$b_2$</th>
<th>$b_1$</th>
<th>$b_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$a_3 \oplus a_2$</td>
<td>$a_3 \oplus a_2 \oplus a_1$</td>
<td>$a_2 \oplus a_0$</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$y$</td>
<td>$x$</td>
<td>$y$</td>
<td>$x$</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$y$</td>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
</tr>
<tr>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
<td>$y$</td>
<td>$x$</td>
</tr>
<tr>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
</tr>
<tr>
<td>$x$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$y$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

pattern.

The linear relationship exploited in this attack is the XOR sum of all bits in the structure. If the XOR sum of all the bits in a particular pattern equals zero, then the pattern is balanced. All patterns described above are balanced, except for the $b_0$-pattern which may or may not be balanced. In order to distinguish between balanced and unbalanced $b_0$-patterns, the pattern $b_0^*$ will be used when the pattern is balanced and $b_0$ otherwise. Examples of text values for the bit patterns $c$, $a_i$ and $b_i$ in a structure of sixteen texts are given in Table 3.1. A specific example is given in Example 3.2.

Example 3.2 Consider a structure of sixteen 8-bit texts. If the texts in the structure are $[63, E3, 67, E7, 73, F3, 77, F7, 23, A3, 27, A7, 33, B3, 37, B7]$, then the structure has the bit-pattern $a_0 a_3 c_2 c_1 c_0$. Table 3.2 depicts this structure in its binary representation. In a traditional integral attack, if this structure is partitioned into two 4-bit sub-blocks, then the structure has the integral property $BB$. This notation does not give much information about the properties of the texts in the structure.
3.1. Bit-Pattern-Based Integral Attack

Table 3.2: Example of 8-bit text values denoted by the pattern $a_0a_3ca_2ca_1cc$

<table>
<thead>
<tr>
<th>$a_0$</th>
<th>$a_3$</th>
<th>$c$</th>
<th>$a_2$</th>
<th>$c$</th>
<th>$a_1$</th>
<th>$c$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

3.1.2 Tracing the Bit Patterns

This section provides an overview on methods for tracing the propagation of bit patterns through linear and nonlinear transformations.

Linear Transformation

The linear transformation is assumed to consist of at least one of these operations: XOR, rotation, shift. Rotations and shifts do not change the bit patterns as these operations just move the patterns to different positions in the structure. If, however, two or more bit patterns are XORed together, then the resulting bit patterns will have the following properties:

- $c \oplus p = p$ for any pattern $p \in \{a_i, b_j\}$.
- $a_i \oplus a_i = c$.
- $p_i \oplus q_j = b_j$ for $i > j$ and $p_i, q_j \in \{a_k, b_l\}$, and $k, l \in \{i, j\}$. If $j = 0$, then the right-hand side of the equation equals $b_0^*$.
- $p \oplus b_0^* = b_0^*$ when $p \neq b_0$, $p \in \{c, a_i, b_0^*\}$.

The first, second and fourth properties are easy to verify. For the third property, an example is given by Example 3.2. The above rules of XOR addition will be
used when analyzing how the bit-patterns in the text block evolve through the linear transformation of a block cipher.

**Example 3.3** Let \( v_0 = [1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0] \), \( v_1 = [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1] \) and \( v_2 = [0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1] \) represent structures of sixteen 1-bit words for the bit patterns \( a_2, a_0 \) and \( b_1 \), respectively. If \( v_0 \oplus v_1 \), which represents \( a_2 \oplus a_0 \), then \( a_2 \oplus a_0 = b_0^g \) because \( v_0 \oplus v_1 = [1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1] \). In another example, if \( v_0 \oplus v_2 \), which represents \( a_2 \oplus b_1 \), then \( a_2 \oplus b_1 = b_1 \) because \( v_0 \oplus v_2 = [1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1] \).

**Nonlinear Transformation**

The nonlinear transformation is assumed to consist only of bijective S-boxes. When the bit patterns pass through a bijective S-box, every output bit of the S-box will have a \( b_i \)-pattern where \( i \) is the smallest index of the input patterns. This is because blocks of \( 2^i \) inputs will all have the same value, and so the output values will also appear in blocks of \( 2^i \) equal values. Example 3.4 illustrates this point.

**Example 3.4** Let \([9, 9, 9, 9, 8, 8, 8, 9, 8, 8, 8, 8, D, D, D, E, C, C, C, C]\) represent a structure of sixteen 4-bit input words to a \( 4 \times 4 \) bijective S-box. The bit pattern for this structure is \( ca_3ca_2 \). Assume that the 4-bit output structure of this S-box is given as follows: \([D, D, D, E, E, E, 0, 0, 0, 0, 7, 7, 7, 7]\). At the output of the S-box, the structure has the bit pattern \( b_2b_2b_2b_2 \) because in every bit position, blocks of \( 2^2 = 4 \) bits have the same value. For clarity, Table 3.3 depicts these input and output structures in their binary representation.

Another useful fact when analyzing the effect an S-box has on input patterns is summed up in Lemma 3.1. This lemma can be helpful when determining whether the balance of a structure is lost through the application of an S-box.

**Lemma 3.1** Consider \( f \) bit sequences, expressed as linear combinations of \( a_i \)-patterns \( l_1, \ldots, l_f \), where \( i \leq g \). Write this using matrix notation as \( M a = 1 \), where \( a = (a_0, \ldots, a_g)^T \) and \( 1 = (l_1, \ldots, l_f)^T \). The different values for the \( f \) bits found in the same position in the sequences lie in an affine space of size \( 2^{\text{rank}(M)} \).

**Proof** Let \( t = f - \text{rank}(M) \). Then there exists \( t \) linearly independent vectors \( v_1, \ldots, v_t \) such that \( v_i M = 0 \), where \( i = 1, \ldots, t \). Since \( a_i \oplus a_i = c \) in our context,
Table 3.3: Example of how a 4-bit text structure denoted by the pattern $ca_3ca_2$ evolves into $b_2b_1b_0b_2$ through a bijective $4 \times 4$ S-box $s$

<p>| Input, $X$ | Output, $s(X)$ |</p>
<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>1001</td>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>D</td>
<td>1101</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>E</td>
<td>1110</td>
</tr>
<tr>
<td>D</td>
<td>1101</td>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>D</td>
<td>1101</td>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>D</td>
<td>1101</td>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>D</td>
<td>1101</td>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>C</td>
<td>1100</td>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>C</td>
<td>1100</td>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>C</td>
<td>1100</td>
<td>7</td>
<td>0111</td>
</tr>
</tbody>
</table>

a 0-row in $M$ corresponds to the constant pattern. This means that all possible values of the $f$ bits lie in an affine formed by the $t$ linear equations given by $v_1, \ldots, v_t$ and $t$ right-hand sides. The size of this space is $2^{f-t}$ and the lemma follows.

**Example 3.5** Assume that one has an $f \times f$ S-box and a structure of $2^g$ texts where $f > g$, and assume the input bits to the S-box are expressed as linear combinations of $a_i$-patterns. Suppose Lemma 3.1 shows that the inputs to the S-box lie in an affine space of dimension smaller than $g$. Then each distinct input value will occur an even number of times, and so each distinct output value will occur an even number of times. Hence the balance will not be lost after the S-box.

### 3.1.3 The Generic Attack

The generic bit-pattern-based integral attack is performed in two phases: distinguishing and key recovery, which are outlined in Algorithm 3.1. The phases are similar to traditional integral attacks, which are given in Section 2.3.6. In the remainder of this chapter, the standard notations defined in Section 2.1.1 will be used.
Distinguishing Phase

This distinguisher is constructed by selecting a structure of plaintexts such that the balanced bit patterns can be propagated through as many rounds as possible. In a basic distinguisher, the structure of plaintexts is selected to have the patterns \( c \) and \( a_i \) only. In the first round, the active S-boxes are set to receive only a single \( a_i \) pattern. The propagation of these balanced patterns is traced until all bits in the structure have the pattern \( b_0 \), i.e., unbalanced. The balance is normally destroyed by the application of the nonlinear transformation after a certain number of rounds.

Let \( \hat{Y}^{\hat{R}(l)}(l) = (Y_{0,l}^{\hat{R}(l)}, Y_{1,l}^{\hat{R}(l)}, \ldots, Y_{m-1,l}^{\hat{R}(l)}) \) and \( \hat{Z}^{\hat{R}(l)}(l) = (Z_{0,l}^{\hat{R}(l)}, Z_{1,l}^{\hat{R}(l)}, \ldots, Z_{m-1,l}^{\hat{R}(l)}) \) denote the \( l \)-th text block of the structure corresponding to the input and output, respectively, of the nonlinear transformation \( S \) in round \( \hat{R} \). The blocks \( Y^{\hat{R}(l)} \) and \( Z^{\hat{R}(l)} \) are partitioned into \( m \) words \( Y_i^{\hat{R}(l)} \) and \( Z_i^{\hat{R}(l)} \), respectively. If the balance of the structure is destroyed after the \( j \)-th S-box in \( S \), then the following linear relationship must hold with probability one:

\[
\bigoplus_{l=0}^{w-1} (Y_{0,l}^{\hat{R}(l)}, Y_{1,l}^{\hat{R}(l)}, \ldots, Y_{m-1,l}^{\hat{R}(l)}) = \bigoplus_{l=0}^{w-1} s^{-1}(Z_{0,l}^{\hat{R}(l)}, Z_{1,l}^{\hat{R}(l)}, \ldots, Z_{m-1,l}^{\hat{R}(l)}) = 0 \quad (3.1)
\]

where \( w \) is the total number of text blocks in the structure.

Key Recovery Phase

In a simple \( R \)-round key recovery attack using an \( \hat{R} \)-round distinguisher where \( \hat{R} = R - 1 \), \( m \) bits of the last round subkey are guessed and a partial decryption of the ciphertext blocks are performed to obtain the value of \((Z_{0,j}^{\hat{R}(l)}, Z_{1,j}^{\hat{R}(l)}, \ldots, Z_{m-1,j}^{\hat{R}(l)})\) for all \( w \) text blocks \((l = 0, 1, \ldots, w)\). Next, a check whether Equation 3.1 holds is performed. If Equation 3.1 holds, then the \( m \)-bit guessed subkey is a possible correct key. Equation 3.1 puts an \( m \)-bit condition on the guess, so the number of possible subkey bit guesses is expected to be reduced by a factor of \( 2^{-m} \). If \( k \) subkey bits are guessed at the same time, then approximately \( \lceil k/m \rceil \) structures are required to identify the correct portion of the subkeys used in the last round. However, in some cases, more structures are required to increase the success rate of the attack. This is demonstrated in Section 3.3.
3.1. Bit-Pattern-Based Integral Attack

**Precomputation (Distinguishing Phase)**
Analyze round function to identify distinguisher;

\[
\begin{align*}
\text{begin} & \\
& \quad (\text{Key Recovery Phase}); \\
& \quad \text{Initialize an array } A[\cdot] \text{ of size } 2^k \text{ bits with all ‘1’s;} \\
& \quad \text{Set } v = 0; \\
& \quad \textbf{while} \text{ number of entries such that } A[v] = 1 \text{ is greater than one } \textbf{do} \\
& \quad \quad \text{Generate a structure of plaintexts that matches distinguisher;} \\
& \quad \quad \text{Encrypt all plaintexts in structure and get corresponding ciphertexts;} \\
& \quad \quad \textbf{for} \text{ subkey guess } v = 0 \text{ to } 2^k - 1 \textbf{ do} \\
& \quad \quad \quad \text{Partially decrypt all ciphertexts using value } v \text{ as partial subkey bits to find the output bits of one S-box in round } r; \\
& \quad \quad \quad \textbf{if} \text{ Equation 3.1 does not hold for subkey } v \textbf{ then} \\
& \quad \quad \quad \quad \text{set } A[v] = 0; \\
& \quad \quad \textbf{end} \\
& \quad \textbf{end} \\
& \quad \textbf{end} \\
& \quad \textbf{end} \\
& \quad \textbf{Output:} \text{ value } v \text{ for which } A[v] = 1 \text{ as correct subkey bits;} \\
\end{align*}
\]

**Algorithm 3.1**: Algorithm for basic attack

**Attack Extensions**
The basic attack can be extended by adding a round at the beginning of the distinguisher. This can be done by assuming the bits in the structure have specific bit-patterns at the input of the second round. The bit patterns in this second round are then traced backwards until they meet the output of the nonlinear transformation in the first round. Since the bit patterns at the output of the active S-boxes are known, one can find the values of the inputs. Next, the value of the subkey bits used in the first round that affect the active S-boxes are guessed. This permits the finding of the structure of plaintexts that will have the specific bit-pattern at the input of the second round, when this guess is correct. If there is a need to guess \(k\) bits in the first round, then one must expect to use at least \(2^k\) structures before a specified bit-patterns is obtained in the second round. This increases the number of required chosen plaintexts, but it may involve a smaller subkey guess than would be needed by adding a round at the end of the distinguisher.
3.2 Applications

The following subsections present the results of the application of the bit-pattern based integral attacks to reduced round versions of bit-based block ciphers Noekeon, Serpent and PRESENT. For each cipher, the state block is illustrated as a block of four rows and 16 or 32 columns. The top row of the block denotes word 0 and the bottom row denotes word 3. The rightmost column corresponds to the least significant bit (bit 0) in each word. The propagation of the distinguisher for each cipher is depicted by highlighting the bit-patterns in the cipher blocks. The key recovery attacks which make use of the distinguisher are then presented. A summary of the complexity of the attacks is given in Table 3.6.

3.2.1 Noekeon

In this section, the bit-pattern-based integral attack on 4- and 5-round versions of Noekeon is described. The description of the cipher is given in Section 2.4.3.

3.5-round Distinguisher

Noekeon reduced to 3.5 rounds can be distinguished from a random permutation by preparing a structure of $2^{16}$ plaintexts which have the following form:

$X^{0(l)} = (X^{0(l)}_0, X^{0(l)}_1, X^{0(l)}_2, X^{0(l)}_3) = (l||D_0, \hat{L}_0(l||D_1), D_2, \hat{L}_0(l||D_3))$

where $D_0, D_1, D_3$ are arbitrary 16-bit constants, $D_2$ is a 32-bit constant and $l = 0, 1, \ldots, 2^{16} - 1$. The bit pattern of this plaintext structure is shown in the first block of Figure 3.3.

In the first round, $V^{0}_{0}$ will become a constant and $V^{0}_{1}$ will cancel the active bits in $X^{0}_1$ and $X^{0}_3$. This leaves the 16 leftmost bits of $Y^{0}_0$ to hold active patterns, i.e. $a_{15}a_{14} \ldots a_{0}$. All other bits hold $c$ patterns. This is shown in the second block of Figure 3.3. The subsequent transformation $S$ will have 16 active S-boxes. Each active S-box receives two distinct inputs which differ only in the leftmost bit. There exists a partial differential\footnote{Refer to Equations 2.6 2.7, 2.8 and 2.9 for clarification.} $8 \rightarrow (w, 3)$ (in hexadecimal) through the S-box with probability one, where $w \in \{0, 1, 2, 3\}$. Consequently, the 16 leftmost bits of both $Z^{0(l)}_2$ and $Z^{0(l)}_3$ have the same $a_i$ pattern as the\footnote{Refer to Table B.1 in Appendix B for the difference distribution table of Noekeon’s S-box.}
3.2. Applications

<table>
<thead>
<tr>
<th>1 3 4 1 3 2 1 0</th>
<th>8 7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 6 5 4 1 3 2 1 0</td>
<td>7 6 5 4 3 2 1 0 0</td>
</tr>
<tr>
<td>7 6 5 4 1 3 2 1 0</td>
<td>7 6 5 4 3 2 1 0 0</td>
</tr>
</tbody>
</table>

$L_1 \circ L_0 \downarrow$

<table>
<thead>
<tr>
<th>8 7 6 5 4 3 2 1 0</th>
<th>0 0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 5 4 3 2 1 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>3 2 1 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>3 2 1 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

$S \downarrow$

<table>
<thead>
<tr>
<th>3 2 1 0 0 0 0 0</th>
<th>0 0 0 0 0 0 0 0</th>
<th>0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 2 1 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>3 2 1 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

$L_1 \circ L_0 \circ L_1^{-1} \circ S \circ L_1 \circ L_0 \circ L_1^{-1} \downarrow$

<table>
<thead>
<tr>
<th>0 0 0 0 0 0 0 0 0</th>
<th>0 0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

$S \downarrow$

Figure 3.3: The 3.5-round bit-pattern-based integral distinguisher for Noekeon.

leftmost bit of the input. The rest of the output bits of the active S-boxes hold a $d_i$ pattern where $0 \leq i \leq 15$. This is depicted in the third block of Figure 3.3.

In the second round, $L_0$ destroys any $c$ pattern from its input structure. This is shown in the fourth block of Figure 3.3. The existence of the partial differential described previously ensures that this occurs with probability one. Every bit of $V_0^1$ and $V_1^1$ contains at least one $a_i$ pattern. After $L_1$, all bit patterns in every column are linearly independent. Therefore, according to Lemma 3.1, there are 16 distinct values repeated $2^{12}$ times at the input of every S-box in the subsequent transformation $S$. The balance is retained after $S$ and all output bits have the $b_i$ ($i \neq 0$) or $b_0^*$ pattern. This is depicted in the fifth block of Figure 3.3. After
Chapter 3. Integral Attack on Bit-Based Block Ciphers

$L_{-1}^{-1}$, the balance of the structure is preserved because the texts in every input column still appear an even number of times.

In the third round, the linear transformations $L_0$ and $L_1$ do not cause any value in any input to an S-box to appear an odd number of times. At the input to $S$, the number of different inputs to each S-box is even and therefore, the number of different outputs is also even. This causes the structure to remain balanced after $S$ and $L_{-1}^{-1}$.

In the fourth round, the balance of the structure is ensured through both $L_0$ and $L_1$. After the application of these linear transformations, the structure holds all $b_0^*$ patterns, as illustrated in the sixth block of Figure 3.3. The balance of the structure is expected to be destroyed after the application of the subsequent $S$. Since one can only positively predict that the balance of the structure is retained before $S$ in the fourth round, this distinguisher covers only 3.5 rounds. This distinguisher can be used to perform key recovery attacks on 4 and 5 rounds of Noekeon using the attack strategy described in Section 3.1.3. The attacks are discussed as follows.

4-round Key Recovery Attack

The key recovery procedure in a 4-round attack is a straightforward process. Once the distinguisher is available, the linear relationship given in Equation 3.1 must hold for $w = 2^{16}$ texts and $\hat{R} = 3$. The following linear equations provide the output bits of the S-boxes in the fourth round:

\[
Z_{0,j}^{3(l)} = X_{0,j}^{4(l)} \oplus \hat{L}_0(X_{1}^{4(l)} \oplus X_{3}^{4(l)})_j \oplus RC_j \oplus K_{0,j} \quad (3.2)
\]

\[
Z_{1,j}^{3(l)} = X_{1,j}^{4(l)} \oplus \hat{L}_0(X_{0}^{4(l)} \oplus X_{2}^{4(l)})_{j-1} \oplus A_{1,j-1} \quad (3.3)
\]

\[
Z_{2,j}^{3(l)} = X_{2,j}^{4(l)} \oplus \hat{L}_0(X_{1}^{4(l)} \oplus X_{3}^{4(l)})_{j-5} \oplus K_{2,j-5} \quad (3.4)
\]

\[
Z_{3,j}^{3(l)} = X_{3,j}^{4(l)} \oplus \hat{L}_0(X_{0}^{4(l)} \oplus X_{2}^{4(l)})_{j-2} \oplus A_{3,j-2} \quad (3.5)
\]

where the bit index $j + n$ is computed modulo 32 ($0 \leq j \leq 31$) and $A_{1,j}$ and $A_{3,j}$ are linear combinations of seven key bits as follows:

\[
A_{i,j} = R(K_0 \oplus K_2)_j \oplus K_{i,j}
\]

\[
= K_{0,j} \oplus K_{2,j} \oplus K_{0,j+8} \oplus K_{2,j+8} \oplus K_{0,j-8} \oplus K_{2,j-8} \oplus K_{i,j}. \quad (3.6)
\]

For the 4-round attack, one needs to guess four bits of key material at the
same time: the bits $K_{0,j}$, $K_{2,j-5}$, $A_{1,j-1}$ and $A_{3,j-2}$. Therefore, approximately $\lceil 4/4 \rceil = 1$ structure is needed to identify a correct guess, however, in practice sometimes two are needed. The guess is repeated 32 times to get 128 bits of key material from the last round key. After the correct values for $K_{0,j}$, $K_{2,j}$, $A_{1,j}$ and $A_{3,j}$ are identified for $j = 0, 1, \ldots, 31$, Equation 3.6 is rearranged and solved to uncover the unknown bits in $K_1$ and $K_3$. The attack requirements are thus $2 \cdot 2^{16} = 2^{17}$ chosen plaintexts and $2 \cdot 2^{16} \cdot 2^4 \cdot 32 = 2^{26}$ partial decryptions.

5-round Key Recovery Attack

In a 5-round attack, the values of the outputs from one S-box in the third round can be obtained by guessing 92 selected bits of information from the subkeys used in rounds 4 and 5. Since Noekeon uses the same subkey in every round, some bits of the subkey that must be guessed in the fourth round overlap with those needed to be guessed in the fifth round. In order to correctly identify all the 92 bits, approximately $\lceil 92/4 \rceil = 23$ different structures have to be used. The remaining $128 - 92 = 36$ bits of subkey material can be found by exhaustive search. The data complexity is therefore $23 \cdot 2^{16} \approx 2^{20.6}$ chosen plaintexts. The time complexity for the attack is $(2^{92} + 2^{88} + \ldots + 2^4 + 1) \cdot 2^{16} + 2^{36} \approx 2^{108.1}$ partial decryptions. Memory is required for storing $2^{92}$ bits indicating possible guesses remaining, thus the memory requirement is $2^{89}$ bytes.

3.2.2 Serpent

In this section, the bit-pattern-based integral attack on 4-, 5- and 6-round versions of Serpent is described. The description of the cipher is given in Section 2.4.4.

3.5-round Distinguisher

Serpent reduced to 3.5 rounds can be distinguished from a random permutation by choosing a structure of $2^{10}$ plaintexts which have the following form:

$$X_0^{(l)} = (X_0^{0(l)}, X_1^{0(l)}, X_2^{0(l)}, X_3^{0(l)}) = (D_0, D_1, l\|D_2, D_3)$$

where $D_0, D_1, D_3$ are 32-bit constants, $D_2$ is a 22-bit arbitrary constant and $l = 0, 1, \ldots, 2^{10} - 1$. The ten leftmost bits of $X_2^{0(l)}$ therefore hold the pattern $a_9 a_8 \ldots a_0$ and the rest of the bits hold a c pattern. The bit pattern of this
plaintext structure is shown in the first block of Figure 3.4. Note that as Serpent uses eight different S-boxes \( s_i \), this distinguisher can start from any round.

In the first round, the transformation \( S_i \) will have 10 active S-boxes. Each active S-box receives a pair of 4-bit input words with a difference of 4 (in hexadecimal). Each input and output word is repeated \( 2^9 \) times, and therefore, the structure after \( S_i \) is balanced. Since each of the eight S-boxes \( s_j \) of Serpent behaves differently to the input difference, the output bits of these S-boxes are denoted as a \( d_i \)-pattern. However, at least two bits of each output word will

---

3 Refer to Section 2.4.4 regarding these eight S-boxes.
4 Refer to Tables B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9 in Appendix B for the difference distribution tables for these different S-boxes.
hold an \(a_i\)-pattern. This is due to one of the design criteria of the S-box \[4\], i.e., one-bit input change results in at least two-bit output change. All other bits have the \(c\) pattern. This is illustrated in the second block of Figure 3.4. The subsequent linear transformation \(L\) does not affect the balance of the structure. The pattern of the output structure is shown in the third block of Figure 3.4.

In the second round, the inputs to each S-Box \(s_{i+1}\) in \(S_{i+1}\) may be a single constant word, or between two and 16 input words. The input words are repeated between \(2^6\) and \(2^{10}\) times. The number of distinct output words is therefore even and this ensures that the structure is balanced after \(S_{i+1}\). The pattern of the output word is given in the fourth block of Figure 3.4. The subsequent transformation \(L\) ensures that the distinct words in every column occur an even number times, or that all words occur an odd number of times. All bits will hold a \(b_0^*\)-pattern except for a few bits which still retain a \(b_1\)-pattern. This is given in the fifth block of Figure 3.4.

In the third round, the input words of each S-box \(s_{i+2}\) in \(S_{i+2}\) are repeated an even or odd number of times. This preserves the balance of the structure until after \(L\). All bits are expected to have the \(b_0^*\)-pattern, which is illustrated in the sixth block of Figure 3.4.

In the fourth round, the application of \(S_{t+3}\) is expected to destroy the balance of the structure. Therefore, all bits hold the \(b_0\)-pattern, which is depicted in the last block of Figure 3.4. Since the balance of the structure can only be predicted before \(S\) with probability one, this distinguisher covers only 3.5 rounds. This distinguisher can be used to perform key recovery attacks on 4, 5 and 6 rounds of Serpent, which are described next. The distinguisher is assumed to start at round \(r = 0\).

### 4-round Key Recovery Attack

In a 4-round attack, the ciphertexts and the output words of \(S\) are separated only by a simple XOR with the last round subkey. Hence only four bits of the last round subkey need to be guessed to verify whether the linear relationship given in Equation 3.1 holds. This 4-round key recovery attack requires \(2 \cdot 2^{10} = 2^{11}\) chosen plaintexts, and it must be repeated 32 times to recover the whole last round subkey. The time complexity is \(2 \cdot 2^{10} \cdot 2^4 \cdot 32 = 2^{20}\) partial decryptions.
This is done by guessing selected bits from $K^2$ obtained from the ciphertext block $X$guisher. Therefore, Equation 3.1 needs to be verified for $\hat{R} = 3$ and $Y^3$ must be obtained from the ciphertext block $X^5$ as follows.

$$Y^3 = S_3^{-1}(Z^3) = S_3^{-1}(L^{-1}(S_4^{-1}(X^5 \oplus K^5) \oplus K^4))$$

This is done by guessing selected bits from $K^4$ and $K^5$. Recall that $Z^3 = L^{-1}(Y^4 \oplus K^4)$ and $Y^4 = S_4^{-1}(X^5 \oplus K^5)$. The following linear equations give the value of the $j$-th bit of the word $Z^3_{i,j}$:

$$Z^3_{0,j} = Y^4_{0,j+18} \oplus Y^4_{1,j+13} \oplus Y^4_{3,j+13} \oplus K^4_{0,j+18} \oplus K^4_{1,j+13} \oplus K^4_{3,j+13},$$

$$Z^3_{1,j} = Y^4_{1,j+1} \oplus Y^4_{0,j+5} \oplus Y^4_{1,j} \oplus Y^4_{2,j+22} \oplus Y^4_{1,j+7} \oplus K^4_{1,j+1} \oplus K^4_{0,j+5} \oplus K^4_{1,j} \oplus K^4_{2,j+22} \oplus K^4_{1,j+7},$$

$$Z^3_{2,j} = Y^4_{2,j+25} \oplus Y^4_{3,j+3} \oplus Y^4_{1,j+3} \oplus K^4_{2,j+25} \oplus K^4_{3,j+3} \oplus K^4_{1,j+3} \oplus K^4_{1,j+3},$$

$$Z^3_{3,j} = Y^4_{3,j+7} \oplus Y^4_{2,j+22} \oplus Y^4_{3,j} \oplus Y^4_{1,j+7} \oplus Y^4_{0,j+3} \oplus Y^4_{1,j+3} \oplus Y^4_{3,j+3} \oplus K^4_{3,j+7} \oplus K^4_{2,j+22} \oplus K^4_{3,j} \oplus K^4_{1,j+7} \oplus K^4_{0,j+3} \oplus K^4_{1,j+3} \oplus K^4_{3,j+3}.$$

In order to obtain $Z^3$, the value of $Y^4$ must first be obtained as follows

$$(Y^4_{0,j}, Y^4_{1,j}, Y^4_{2,j}, Y^4_{3,j}) = s_4^{-1}(X^5_{0,j} \oplus K^5_{0,j}, X^5_{1,j} \oplus K^5_{1,j}, X^5_{2,j} \oplus K^5_{2,j}, X^5_{3,j} \oplus K^5_{3,j}).$$

To compute the output of $s_3^{-1}$ at bit $j = 0$, one needs to guess 36 bits of $K^5$ and 11 bits of $K^4$ (47 bits in total). Repeating this for bit $j = 1$, then there is a need to guess 32 bits of $K^5$ and 10 bits of $K^4$ (42 bits in total). If this process is continued further and repeated for bit $j = 2$, then 20 bits of $K^5$ and 10 more bits of $K^4$ (30 bits in total) must be guessed. At this point, a total of 11 + 10 + 10 = 31 bits of $K^4$ and 36 + 32 + 20 = 88 bits of $K^5$ have been obtained. The remaining 128 − 88 = 40 bits of $K^5$ can then be found by exhaustive search.

<table>
<thead>
<tr>
<th>$K^4_{0,j}$</th>
<th>$K^4_{1,j}$</th>
<th>$K^4_{2,j}$</th>
<th>$K^4_{3,j}$</th>
<th>$K^5_{0,j}$</th>
<th>$K^5_{1,j}$</th>
<th>$K^5_{2,j}$</th>
<th>$K^5_{3,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit $j = 0$</td>
<td>5, 18</td>
<td>0, 1, 13</td>
<td>22, 25</td>
<td>0, 3, 7, 13</td>
<td>0, 1, 3, 5, 7, 13</td>
<td>18, 22, 25</td>
<td></td>
</tr>
<tr>
<td>Bit $j = 1$</td>
<td>6, 19</td>
<td>1, 2, 14</td>
<td>23, 26</td>
<td>1, 4, 8, 14</td>
<td>1, 2, 4, 6, 8, 14</td>
<td>19, 23, 26</td>
<td></td>
</tr>
<tr>
<td>Bit $j = 2$</td>
<td>7, 20</td>
<td>2, 3, 15</td>
<td>24, 27</td>
<td>2, 5, 9, 15</td>
<td>2, 3, 5, 7, 9, 15, 20, 24, 27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3.5: Bit patterns of $Z_0$ in the 6-round attack

<table>
<thead>
<tr>
<th>Word</th>
<th>Bit Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_0^0$</td>
<td>c c c c . . . c c c c c</td>
</tr>
<tr>
<td>$Z_1^0$</td>
<td>c c c c . . . a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0</td>
</tr>
<tr>
<td>$Z_2^0$</td>
<td>a_2 a_1 a_0 c . . . c c c a_9 a_8 a_7 a_6 a_5 a_4 a_3</td>
</tr>
<tr>
<td>$Z_3^0$</td>
<td>c c c c . . . c a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0</td>
</tr>
</tbody>
</table>

Table 3.4 lists the bit positions $\hat{j}$ of the subkeys $K^4_{i,j}$ and $K^5_{i,j}$ that need to be guessed.

The three sets of guesses (for $j = 0, 1, 2$) are performed for the same structure and can be done in parallel. Out of these three sets of guesses, there is a need to guess a maximum of 47 bits of subkey material at any one time. In order to correctly identify all subkey bits, about $\lceil 47/4 \rceil = 12$ structures are needed, which equates to $12 \cdot 2^{10} \approx 2^{13.6}$ chosen plaintexts. The time requirement is approximately $(2^{47} + 2^{43} + \ldots + 1) \cdot 2^{10} \cdot 3 + 2^{40} \approx 2^{58.7}$ partial decryptions. The required memory for storing the possible subkey candidates is $2^{47}/2^3 = 2^{44}$ bytes.

### 6-round Key Recovery Attack

The 5-round attack can be extended to a 6-round key recovery attack by adding one round at the beginning. The 10 most significant bits of $X^1_2$ must hold the pattern $a_9 a_8 \ldots a_0$ and the rest of the bits must hold a c pattern. If the active bits in $X^1_2$ are traced backwards until the input of $L$ in the first round (Round 0), the bits assume the pattern shown in Table 3.5 where $c \ldots c$ denotes 19 consecutive bits which have the pattern c. There are 13 active S-boxes in the first round, hence $13 \cdot 4 = 52$ bits of $K^0$ must be guessed to find a plaintext structure that will evolve into the desired pattern in $X^1$ when the guess is right. The attack requires $\lceil (52 + 47)/4 \rceil \times 2^{52} \approx 2^{56.64}$ chosen plaintexts. The time complexity is about $(2^{99} + 2^{95} + \ldots + 1) \cdot 2^{10} \approx 2^{109.09}$ partial decryptions for a single structure. The memory requirement is the same as in the 5-round attack.

### 3.2.3 PRESENT

In this section, the bit-pattern-based integral attack on 4-, 5-, 6- and 7-round versions of PRESENT is described. The description of the cipher is given in Section 2.4.5.
3.5-round Distinguisher

PRESENT reduced to 3.5 rounds can be distinguished from a random permutation by constructing a structure of $2^4$ chosen plaintexts which have the following form:

$$(X_0^{0(l)}, X_1^{0(l)}, X_2^{0(l)}, X_3^{0(l)}) = (D_0, D_1, D_2, D_3 || l)$$

where $D_0, D_1, D_2$ are arbitrary 16-bit constants, $D_3$ is an arbitrary 12-bit constant and $l = 0, 1, \ldots, 15$. The bit pattern for this plaintext structure is shown in the first block of Figure 3.5.

In the first round, each of the four rightmost active S-boxes in the nonlinear
transformation $S$ receives two different input words repeated eight times. The rest of the S-boxes receive only a single constant value repeated 16 times. The output bits of the four rightmost S-boxes assume the pattern $d_i d_i d_i a_i$ since there is a partial differential $\delta^1 \rightarrow (w, 1)$ (in hexadecimal) occurring with probability one where $w \in \{1, 3, 4, 6\}$. The bit pattern for this output structure is given in the second block of Figure 3.5. The subsequent transformation $L$ spreads the $a_i$ and $d_i$ bit patterns to four S-boxes in the next round. This is illustrated in the third block of Figure 3.5.

In the second round, there are at most four active S-boxes in $S$. The S-box instances are 0, 4, 8 and 12. S-box 0 receives the pattern $a_3 a_2 a_1 a_0$ which represents all possible 4-bit values. S-boxes 4, 8 and 12 receive the pattern $d_3 d_2 d_1 d_0$. The inputs to the other 12 S-boxes have the pattern $c$. The input and output values of the active S-boxes are repeated either once or an even number of times, hence, the structure is balanced after $S$. The pattern of the output bits for these active S-boxes is $b_0^5$, which is depicted in the fourth block of Figure 3.5. The subsequent transformation $L$ spreads the bits such that each S-box in $S$ in the third round has the input bit pattern $c c c b_0^*$. This is illustrated in the fifth block of Figure 3.5.

In the third round, since only one bit position is non-constant, all S-boxes in $S$ receive at most two different input words. Recall that in the second round, the output of S-box 0 consists of all possible 4-bit values. The subsequent transformation $L$ ensures that the number of repetitions for the different input words for S-boxes 0, 4, 8 and 12 in the current (third) round is exactly eight. The output bits of all S-boxes at this point hold the pattern $b_0^*$ and the structure remains balanced after $S$. The subsequent $L$ ensures this balanced state into the next round.

In the fourth round, since the balance of the structure can only be determined before $S$ with probability one, the distinguisher covers only 3.5 rounds. This distinguisher can be used to perform key recovery attacks on 4, 5, 6 and 7 rounds of Present, which will be discussed in the following sections.

**4-round Key Recovery Attack**

Recall that after the last round of Present, there is a key addition transformation. Therefore, only four bits of round subkey $K^4$ need to be guessed at a

\[ \text{Refer to Table B.10 in Appendix B for the difference distribution table for this S-box.} \]
time to utilize Equation 3.1. The guesses are then performed for the outputs of the remaining fifteen S-boxes in the last round. In summary, the 4-round key recovery attack needs $2 \cdot 2^4 = 2^5$ chosen plaintexts and $2 \cdot 2^4 \cdot 16 \cdot 2^4 = 2^{13}$ partial decryptions.

**5-round Key Recovery Attack**

In a 5-round attack, one more round is appended to the fourth round. The attacker needs to guess four bits of $K^4$ and an additional $4 \cdot 4 = 16$ bits of key material from $K^5$ (20 bits of guesses in total). Therefore, approximately $\lceil (4 + 16)/4 \rceil = 5$ structures are needed to identify the correct guess. The attack is repeated four times (each time using a different S-box in the fourth round) to obtain $4 \cdot 4 = 16$ bits of $K^4$ and the entire $4 \cdot 16 = 64$ bits of $K^5$. The number of chosen plaintexts needed is $5 \cdot 2^4 \approx 2^{10}$, and the time complexity is $5 \cdot 2^4 \cdot (2^{20} + 2^{16} + \ldots + 1) \cdot 4 \approx 2^{24.4}$ partial decryptions. The memory requirement is small.

**6-round Key Recovery Attack**

A 6-round attack can be launched by adding one round at the beginning to construct a 4.5-round distinguisher. The plaintexts are chosen such that the inputs to the second round assume the input pattern of the 3.5-round distinguisher described above. There are four active S-boxes in the first round, and hence 16 bits of $K^0$ need to be guessed. In total, there are $16 + 4 + 16 = 36$ bits of subkey guesses. This 6-round attack would require $2^{16} \cdot 5 \cdot 2^4 \approx 2^{22.4}$ chosen plaintexts and $2^{16} \cdot 2^{25.7} \approx 2^{41.7}$ partial decryptions. The memory complexity is still small.

**7-round Key Recovery Attack**

One can extend the attack to seven rounds by adding another round at the end. However, this attack is only better than exhaustive search for the 128-bit key variant of Present. In a 7-round attack, the entire 64 bits of $K^7$ need to be guessed. A closer examination of the key schedule for 128-bit keys reveals that 3 bits of $K^6$ and 58 bits of $K^5$ are given from guessing all of $K^7$. These known bits overlap with one of the 16 bits needed from $K^6$ and three of the four bits needed from $K^5$, so in total one needs to guess $(16 - 1) + (4 - 3) + 64 = 80$ bits of round subkey material. In summary, the 7-round key recovery attack requires
20·2^{16}·2^4 \approx 2^{24.3} \text{ chosen plaintexts and } (2^{80} + 2^{76} + \ldots + 1)·2^4·2^{16} \approx 2^{100.1} \text{ partial decrypts}. A \text{ total of } 2^{80} \text{ bits are required to keep track of possible values for the 80 key bits, so the memory complexity for this attack is } 2^{80}/2^3 = 2^{77} \text{ bytes.}

### 3.3 Experimental Results

Note that all 4-round key recovery attacks presented in Section 3.2 are practical to implement. Refer to Table 3.6 for the complexities. As proof of concept, a simple C program was developed to implement these attacks. The implementation was performed on a single desktop computer running a 3 GHz Intel Pentium 4 processor with 1 GB of RAM.

#### 3.3.1 Format of Experiments

The experiments were conducted as follows. For each block cipher, the attack was run using different number of structures. For each of the different sets of structures, $2^{10}$ trials of a slightly modified Algorithm 3.1 were run by removing the while loop. The aims of this experiment were to calculate: the data complexity, the average attack timings (in seconds), the average number of possible correct subkey suggestions and the success rate of the attack for different numbers of structure sets.

In the precomputation stage, two 16-element arrays denoted by $A[]$ and $B[]$ were initialized to all ‘1’s and to all ‘0’s, respectively. Next, a counter was then initialized to $n = 0$ to count the total number of possible correct subkey

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Rounds</th>
<th>Complexity</th>
<th>Data</th>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noekeon</td>
<td>4</td>
<td>$2^{17}$ CP</td>
<td>$2^{17}$</td>
<td>$2^{26}$</td>
<td>small $2^{89}$ bytes</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>$2^{20.6}$ CP</td>
<td>$2^{108.1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serpent</td>
<td>4</td>
<td>$2^{11}$ CP</td>
<td>$2^{20}$</td>
<td></td>
<td>small $2^{44}$ bytes</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>$2^{13.6}$ CP</td>
<td>$2^{58.7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>$2^{56.64}$ CP</td>
<td>$2^{13.09}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Present</td>
<td>4</td>
<td>$2^{6}$ CP</td>
<td>$2^{13}$</td>
<td></td>
<td>small $2^{77}$ bytes</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>$2^{6.4}$ CP</td>
<td>$2^{25.7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>$2^{22.4}$ CP</td>
<td>$2^{41.7}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>$2^{24.3}$ CP</td>
<td>$2^{100.1}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
begin
    Initialize an array $B[]$ of 16 counters set to all '0's;
    Set $n = 0$;
    for number of trials $i = 1$ to $2^{10}$ do
        Generate a random master key;
        while number of structures is less than 2, 4 or 8 do
            Generate a random structure of plaintexts;
            Encrypt all plaintexts in structure with 4-round cipher;
            for partial subkey guess $v = 0$ to $F$ do
                Partially decrypt ciphertexts using value $v$ as partial
                subkey;
                if Equation 3.1 does not hold for subkey $v$ then
                    Set $A[v] = 0$;
                end
            end
        end
        for subkey value $v = 0$ to $F$ do
            if $A[v] = 1$ then
                Increment $m = m + 1$;
            end
        end
        Increment $B[m] = B[m] + 1$;
        Add $n = n + m$;
    end
end

// $m$ counts the number of suggested subkeys
Set $m = 0$;
for subkey value $v = 0$ to $F$ do
    if $A[v] = 1$ then
        Increment $m = m + 1$;
    end
Increment $B[m] = B[m] + 1$;
Add $n = n + m$;
end

Output: average number of suggestions per trial, $n/2^{10}$;
Output: success rate, $B[1]/2^{10}$;

Algorithm 3.2: Algorithm for conducting the bit-pattern-based integral at-
tack experiments

suggestions. After all the arrays and counters were set, the trials were ready to
begin.

In the online stage, for each trial, a random master key and a random struc-
ture of plaintexts were generated. Using this master key, the plaintexts were
encrypted to obtain the corresponding ciphertexts. Next, for every 4-bit guess $v$
of the selected word in the last round subkey, the affected ciphertext words were
decrypted to obtain the value of the input words of the affected S-box in the
penultimate round. Then, these input words were checked to see whether they
agreed with the linear relationship given in Equation 3.1. If they did not agree,
then the element $A[v]$ was set to zero. After all subkey values had been guessed,
3.3. Experimental Results

Table 3.7: Experimental results of bit-pattern based integral attack to Noekeon, Serpent and PRESENT reduced to 4 rounds.

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Number of Structures</th>
<th>Data Complexity</th>
<th>Average Time (seconds)</th>
<th>Average Number of Suggestions</th>
<th>Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noekeon</td>
<td>2</td>
<td>$2^{17}$</td>
<td>$2^{-1.33}$</td>
<td>$2^{0.52}$</td>
<td>66.02</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>$2^{18}$</td>
<td>$2^{-0.44}$</td>
<td>$2^{0.08}$</td>
<td>94.04</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>$2^{19}$</td>
<td>$2^{0.60}$</td>
<td>$2^{0.01}$</td>
<td>99.02</td>
</tr>
<tr>
<td>Serpent</td>
<td>2</td>
<td>$2^{11}$</td>
<td>$2^{-4.73}$</td>
<td>$2^{0.14}$</td>
<td>89.75</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>$2^{12}$</td>
<td>$2^{-3.74}$</td>
<td>$2^{0.03}$</td>
<td>99.80</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>$2^{13}$</td>
<td>$2^{-2.72}$</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>PRESENT</td>
<td>2</td>
<td>$2^{5}$</td>
<td>$2^{-7.34}$</td>
<td>$2^{3.02}$</td>
<td>16.89</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>$2^{6}$</td>
<td>$2^{-6.23}$</td>
<td>$2^{2.24}$</td>
<td>35.84</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>$2^{7}$</td>
<td>$2^{-5.39}$</td>
<td>$2^{1.04}$</td>
<td>63.57</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>$2^{8}$</td>
<td>$2^{-4.26}$</td>
<td>$2^{0.22}$</td>
<td>89.45</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>$2^{9}$</td>
<td>$2^{-3.31}$</td>
<td>$2^{0.01}$</td>
<td>99.22</td>
</tr>
</tbody>
</table>

while using the same master key, a different set of structures was generated. This process was repeated until the number of required structures was reached. After all structures were generated, the number of elements in the array $A[i]$ such that $A[v] = 1$ were counted. If the total count was $m$, then the counter $B[m]$ was incremented. This process was repeated until the required number of trials was reached.

After all the trials were completed, the following values were calculated: average number of correct subkey suggestions, average time to complete the attack per trial and the success rate of the attack. The success rate denotes the ratio between the number of trials that suggest only one correct subkey and the total number of trials. These steps are outlined in Algorithm 3.2.

In these experiments, the attack was run using sets of two, four and eight different structures for all block ciphers. However, for PRESENT, the experiments were conducted using two additional sets of structures: 16 and 32. These additional sets were included in the investigation to enable identification of the approximate number of structures required to conduct attacks with high success rates (greater than 90%). Each structure was composed of $2^{16}$ plaintext blocks for Noekeon, $2^{10}$ for Serpent and $2^{4}$ for PRESENT. The results of the experiments are given in Table 3.7 for each block cipher and for the different sets of structures.

---

Refer to Section 3.2 to recall why these many plaintexts are used.
3.3.2 Discussion of Results

Recall that for all 4-round attacks, the theoretical number of structures required to launch the attacks is two. From Table 3.7 if two structures are used in the attack, then the success rate was 66.02% for Noekeon, 89.75% for Serpent and 16.89% for Present. The experimental results therefore confirm the theoretical results, albeit with different success rates for each cipher. If more structures are used, then the success rate of the attack will increase. In particular, a 100% success rate was achieved for Serpent when eight structures were used in the attack.

The different success rates for the attack might be due to the amount of available information (the plaintexts) in the structure and the linear transformation in the last round. In the case of using two different structures, the attack on Noekeon uses the highest number of chosen plaintexts with $2^{17}$. This is followed by Serpent with $2^{11}$ and Present with $2^5$. For Noekeon and Serpent, much more information is available to discard wrong subkey bits compared to Present. This might be the reason why the attack on Present needs more structures than the attacks on Noekeon and Serpent to have a high success rate. However, the attacks on Noekeon have lower success rates than the attacks on Serpent even though the former uses more plaintexts than the latter. This is because, in Noekeon, due to the linear transformation $L_0$, the output bits of one S-box in the last round depend on many ciphertext and subkey bits. In contrast, in Serpent, the output bits of any one S-box in the last round depend only on four ciphertext and four subkey bits.

Note that the timings recorded for the experiments may not depict the actual running time of the attack. Furthermore, no programming tricks were used to optimize the C code. The sole purpose of the experiments was to verify the theoretical results. Note that the time complexity of the theoretical attack is measured based on the number of operations and this is not the same as the timings recorded in the experiments.

3.4 Discussion

In the Noekeon specification, it is stated that there are no 4-round differential trails with a predicted propagation ratio above $2^{-48}$. A propagation ratio is the fraction of input pairs with a fixed difference that propagates into a fixed out-
3.5 Related Work

Knudsen and Wagner [112] mentioned the applicability of the integral attack to bit-based block ciphers. The attack is demonstrated on the Data Encryption
Standard (DES). However, no generic methods were proposed to describe the attack. Lucks \cite{129} also attacked Twofish, which is not a purely byte-based cipher, with integral cryptanalysis. In Piret’s thesis \cite{154 pg 79-82}, the construction of an integral distinguisher for Serpent was discussed. The distinguisher constructed by Piret, however, does not occur with probability one and the number of rounds of the distinguisher was not explicitly mentioned. In comparison, the 3.5-round integral distinguisher identified for Serpent in this chapter occurs with probability one.

In another work, Biryukov and Shamir \cite{34} show how to attack a generic cipher structure which consists of nonlinear and linear transformations which are unknown. The technique, called the multiset attack, makes use of several multiset properties. These properties take into account whether the multiset: (1) contains arbitrary repetitions of a single value; (2) takes on all possible values; (3) contains values which occur an even number of times; (4) has an XOR sum that equals 0; (5) has either property (2) or (3). Therefore, there are some similarities to the notations described in this work.

### 3.6 Summary and Conclusion

This chapter presents new methods to apply the traditional word-based integral attack to bit-based block ciphers. It differs from classical integral cryptanalysis in that the order of the texts in a structure becomes important, and gives the cryptanalyst a more refined notation for the texts in the structure. This information allows an attacker to gain a detailed analysis of the individual bit that propagates through the rounds. This is especially useful in analyzing the attack on ciphers that have bit-based round functions. The application of the bit-pattern based integral attacks to the bit-based block ciphers Noekeon, Serpent and PRESENT have been demonstrated in this chapter. A summary of these attacks is given in Table 3.6.

Over a small number of rounds, bit-pattern based integral cryptanalysis of the three ciphers studied here is comparable to differential cryptanalysis in time complexity, but requires, in general, much less chosen plaintext. However, differential cryptanalysis is more readily extended to a greater number of rounds, whereas integral cryptanalysis cannot be extended beyond a certain point. Even though the attacks presented in this chapter do not pose a serious threat to the
full version of the ciphers, they clearly demonstrate that integral attacks can be applied to bit-based ciphers. Furthermore, this is the first time that a generic method has been proposed to apply integral attacks to bit-based block ciphers.
Chapter 4

Algebraic Analysis of LEX-AES

LEX-AES [27], which was described in Section 2.4.7, is a stream cipher based on the AES block cipher. This chapter presents the results of the algebraic analysis of nonlinear and linear relationships existing in LEX-AES. LEX-AES is an example of using the block cipher AES in the generic LEX technique proposed by the designers of LEX [27]. Since the round function of the AES is used in the keystream generator of LEX-AES, algebraic interaction between the keystream and internal state as a result of using the round function is analyzed.

One of the motivations for exploring algebraic relations in LEX-AES is because it inherits the rich algebraic structure of the AES. In particular, the AES has been described as a system of continued fractions over \( GF(2^8) \) [77]. It has also been studied under the so-called XSL attack using a system of equations over \( GF(2) \) [55]. Furthermore, simple multivariate quadratic equations over \( GF(2^8) \) can be derived by embedding the AES in the larger cipher called the Big Encryption System (BES) [136]. It is therefore natural to study LEX-AES from the perspective of algebraic attacks.

In this chapter, the results of investigation into both LEX-AES and LEX-BES are presented. LEX-BES is a variant of LEX-AES where the underlying AES block cipher is embedded in the BES [136]. In both cases, the analysis involves forming a system of equations that links the keystream, internal state and secret key bytes. A known plaintext attack is assumed, so the attacker has access to a sequence of known plaintext and corresponding keystream.

For LEX-AES, it is shown for the first time that a very small system of
Chapter 4. Algebraic Analysis of LEX-AES

21 equations in 17 unknown bytes can be directly constructed. The amount of keystream required to form the equations is only 36 bytes, which is very small and reasonably practical to obtain. For LEX-BES, experiments were conducted to investigate algebraic cryptanalysis on small scale variants of LEX-BES using the MAGMA computational algebra package \[38\]. This follows a similar work done by Cid, Murphy and Robshaw that investigated algebraic cryptanalysis on small scale variants of the BES \[46\]. The difference between LEX-BES and BES is that due to the way the keystream is extracted, the number of unknowns in LEX-BES equations is fewer than the number in BES. As far as the author knows, these attempts are the first at creating solvable equation systems for stream ciphers based on the LEX method.

This chapter is organized as follows. Section 4.1 presents some preliminaries that provide the context for subsequent sections. Straightforward equations arising from LEX-AES are discussed in Section 4.2. Section 4.3 explores the equation system arising from LEX-BES. A discussion of our findings is given in Section 4.4. Section 4.5 presents a summary of this chapter and some conclusions.

4.1 Preliminaries

As noted in Section 2.3.9, there are two main phases in an algebraic attack: deriving a system of equations and solving the system. Both of these phases are discussed in this chapter. Note that the system of equations constructed contains terms which represent both linear and nonlinear relationships within the cipher.

The algebraic approach presented in this chapter assumes that the attacker knows some plaintext and ciphertext pairs. For a binary additive stream cipher, this is equivalent to knowing the keystream output. In LEX-AES, this also means that portions of the internal state are known because the keystream is extracted directly from the internal state. Under the known keystream assumption, these keystream bytes are considered as known constants. In the LEX-AES system of equations, the unknown portions of the internal state and subkeys are considered as variables.

In this chapter, the terms round and iteration are used interchangeably. This is because the keystream generation of LEX-AES can be seen as repeated invocation of the same ten rounds of the AES using the same ten subkeys, provided

\[1\]Refer to Section 2.2 for a brief description of a stream cipher.
that the secret key remains unchanged. This means that the subkey used in every 10-th iteration is the same. Therefore, iteration \( t \) can be viewed as round \( t - \lfloor t/10 \rfloor \cdot 10 \) of the \( \lfloor t/10 \rfloor \)-th invocation. For ease of description, a set of different notations are used in this chapter, which are different from those given in Section 2.1.1.

### 4.2 Forming Equations to Describe LEX-AES

In order to construct the equation system for LEX-AES, one needs to obtain the output keystream from nine consecutive iterations. This amounts to \( 9 \times 4 = 36 \) bytes of keystream. The keystream is then described as a function of the 16-byte internal state at a particular fixed iteration. The system of equations contains relationships in the keystream generation and key schedule algorithms described in the following subsections.

#### 4.2.1 Keystream Generation Equations

Recall from Section 2.4.7 that the 4-byte keystream at iteration \( t \) is extracted from the 16-byte internal state \( X^t \). For the sake of discussion, \( X^4 \) is set as the internal state on which all equations are based. The internal state \( X^4 \) is
Chapter 4. Algebraic Analysis of LEX-AES

referred to as the middle internal state and $t = 4$ as the middle iteration. For ease of notation, the constants and variables are labelled as follows. The 12 variables in $X^4$ are denoted as $(x_0, \ldots, x_{11})$. The constants (keystream bytes) and subkey variables at iteration $t$ are denoted by $(c_{4t}, c_{4t+1}, c_{4t+2}, c_{4t+3})$ and $K^t = (k_{16t}, \ldots, k_{16t+15})$, respectively. Figure 4.1 visually depicts the positions of these constants and variables, and Figure 4.2 shows the corresponding subkeys. In Figure 4.1, the top left is the internal state at iteration $t = 0$ and the bottom right is the internal state at iteration $t = 8$. The values for the subkey variables and the temporary variables $p_j$, $q_j$, $r_j$, $s_j$, $t_j$, $u_j$ are given in Appendices C.1.1 and C.1.2. These temporary variables are described in terms of only the $x_j$’s and the subkey variables.

The system of equations is built by describing all constants (except the ones in the middle) in terms of the 12 state variables at the middle internal state. There are 32 constants at the outer iterations and thus, there are 32 equations expressed in terms of 12 state variables. Figure 4.3 shows an example of building equations describing two constants: one before and one after the middle iteration. It clearly shows that a constant at iteration $i$ depends on the values of one subkey byte and four state bytes at iteration $i - 1$. However, a constant at iteration $i$ depends on the values of four subkey bytes and four state bytes at iteration $i + 1$.

One now has a set of equations describing 32 out of the 36 bytes of known keystream. The system initially contains 32 equations, expressed in terms of 12

---

2Note that similar equations can be generated when a different middle iteration is used.
4.2. Forming Equations to Describe LEX-AES

One round backward
\[ c_{12} = S^{-1}[E_{x_2} \oplus B_{x_1} \oplus D_{x_4} \oplus 9_{x_5} \oplus E_{k_{12}} \oplus B_{k_{55}} \oplus D_{k_{54}} \oplus 9_{k_{53}}] \]

One round forward
\[ c_{20} = 2S[x_2] \oplus 3S[x_6] \oplus S[x_{10}] \oplus S[x_1] \oplus k_{68} \]

Figure 4.3: Example of forming equations to describe two constants: one before and one after the middle iteration. Known keystream bytes (constants) are denoted in gray.

state variables and 108 subkey variables. The 12 state variables are denoted by \(x_0, x_1, \ldots, x_{11}\). The subkey variables consist of 8 bytes from \(K^0\), \(6 \times 16 = 96\) bytes from \(K^1, \ldots, K^6\) and 4 bytes from \(K^7\) (for a total of \(8 + 96 + 4 = 108\) subkey variables). Note that if the 12 state variables in the middle are not fixed, the number of state variables is \(8 + 60 + 8 = 76\). In this case, the additional state variables come from the temporary variables \(p_j, q_j, r_j, s_j, t_j, u_j\). However, in the case being discussed here, these temporary state variables are described in terms of \(x_j\) (\(0 \leq j \leq 11\)). A process to reduce the number of subkey variables is explained in Section 4.2.2.

Next, there is a need to eliminate some state variables. This can be done using the following eight equations, which describe the constant at iteration \(t = 3\) and
t = 5:

\[ c_{20} = \Theta(x_2, x_6, x_{10}, x_1) \oplus k_{68} \]  
(4.1)
\[ c_{21} = \Theta(x_{10}, x_1, x_2, x_6) \oplus k_{70} \]  
(4.2)
\[ c_{22} = \Theta(x_8, x_0, x_4, x_7) \oplus k_{76} \]  
(4.3)
\[ c_{23} = \Theta(x_4, x_7, x_8, x_0) \oplus k_{78} \]  
(4.4)
\[ s[c_{12}] = \Pi(x_2, x_3, x_4, x_5) \oplus \Pi(k_{52}, k_{53}, k_{54}, k_{55}) \]  
(4.5)
\[ s[c_{13}] = \Pi(x_{10}, x_{11}, x_8, x_9) \oplus \Pi(k_{62}, k_{63}, k_{60}, k_{61}) \]  
(4.6)
\[ s[c_{14}] = \Pi(x_8, x_9, x_{10}, x_{11}) \oplus \Pi(k_{60}, k_{61}, k_{62}, k_{63}) \]  
(4.7)
\[ s[c_{15}] = \Pi(x_4, x_5, x_2, x_3) \oplus \Pi(k_{54}, k_{55}, k_{52}, k_{53}) \]  
(4.8)

where the functions \( \Theta(\cdot) \) and \( \Pi(\cdot) \), which represent MixColumns and its inverse, respectively, are given by:

\[ \Theta(z_0, z_1, z_2, z_3) = 2s[z_0] \oplus 3s[z_1] \oplus s[z_2] \oplus s[z_3], \]
\[ \Pi(z_0, z_1, z_2, z_3) = Ez_0 \oplus Bz_1 \oplus Dz_2 \oplus 9z_3. \]

The previous eight equations can be used to eliminate eight state variables using substitution. Assume that it is required to eliminate the following variables: \( x_0, x_1, x_3, x_5, x_6, x_7, x_9 \) and \( x_{11} \). These variables can be described as:

\[ x_6 = s^{-1}[\theta(s[x_2], s[x_{10}], k_{68}, k_{70}, c_{20}, c_{21})] \]  
(4.9)
\[ x_1 = s^{-1}[\theta(s[x_{10}], s[x_2], k_{70}, k_{68}, c_{21}, c_{20})] \]  
(4.10)
\[ x_0 = s^{-1}[\theta(s[x_8], s[x_4], k_{76}, k_{78}, c_{22}, c_{23})] \]  
(4.11)
\[ x_7 = s^{-1}[\theta(s[x_4], s[x_8], k_{78}, k_{76}, c_{23}, c_{22})] \]  
(4.12)
\[ x_3 = \pi(x_2, x_4, k_{52}, k_{53}, k_{54}, s[c_{12}], s[c_{15}]) \]  
(4.13)
\[ x_{11} = \pi(x_{10}, x_8, k_{62}, k_{63}, k_{60}, s[c_{13}], s[c_{14}]) \]  
(4.14)
\[ x_9 = \pi(x_8, x_{10}, k_{60}, k_{61}, k_{62}, s[c_{14}], s[c_{13}]) \]  
(4.15)
\[ x_5 = \pi(x_4, x_2, k_{54}, k_{55}, k_{52}, s[c_{15}], s[c_{12}]) \]  
(4.16)

where the functions \( \theta(\cdot) \) and \( \pi(\cdot) \) are defined as follows:

\[ \theta(z_0, z_1, z_2, z_3, z_4, z_5) = 47z_0 \oplus CB(z_1 \oplus z_3 \oplus z_5) \oplus 46(z_2 \oplus z_4), \]
\[ \pi(z_0, z_1, z_2, z_3, z_4, z_5, z_6) = 47(z_0 \oplus z_2) \oplus CB(z_1 \oplus z_4) \oplus z_3 \oplus 44z_5 \oplus C9z_6. \]
4.2. Forming Equations to Describe LEX-AES

The step-by-step procedure to obtain Equations 4.9-4.16 is outlined below.

1. Describe $x_6$ in Eq. 4.1 in terms of $x_1, x_2, x_{10}$.
2. Describe $x_1$ in Eq. 4.2 in terms of $x_2, x_6, x_{10}$.
3. Substitute $x_6$ in Step 2 with $x_6$ in Step 1. Now $x_1$ is described only in terms of $x_2, x_{10}$ (Refer to Eq. 4.9).
4. Substitute $x_1$ in Step 1 with $x_1$ in the previous step. Now $x_6$ is described only in terms of $x_2, x_{10}$ (Refer to Eq. 4.10).
5. Describe $x_0$ in Eq. 4.3 in terms of $x_4, x_7, x_8$.
6. Describe $x_7$ in Eq. 4.4 in terms of $x_0, x_4, x_8$.
7. Substitute $x_0$ in Step 6 with $x_0$ in Step 5. Now $x_7$ is described only in terms of $x_4, x_8$ (Refer to Eq. 4.12).
8. Substitute $x_7$ in Step 5 with $x_7$ in the previous step. Now $x_0$ is described only in terms of $x_4, x_8$ (Refer to Eq. 4.11).
9. Describe $x_3$ in Eq. 4.5 in terms of $x_2, x_4, x_5$.
10. Describe $x_{11}$ in Eq. 4.6 in terms of $x_8, x_9, x_{10}$.
11. Describe $x_9$ in Eq. 4.7 in terms of $x_8, x_{10}, x_{11}$.
12. Describe $x_5$ in Eq. 4.8 in terms of $x_2, x_3, x_4$.
13. Substitute $x_5$ in Step 9 with $x_5$ in Step 12. Now $x_3$ is described only in terms of $x_2, x_4$ (Refer to Eq. 4.13).
14. Substitute $x_9$ in Step 10 with $x_9$ in Step 11. Now $x_{11}$ is described only in terms of $x_8, x_{10}$ (Refer to Eq. 4.14).
15. Substitute $x_{11}$ in Step 11 with $x_{11}$ in Step 14. Now $x_9$ is described only in terms of $x_8, x_{10}$ (Refer to Eq. 4.15).
16. Substitute $x_3$ in Step 12 with $x_3$ in Step 13. Now $x_5$ is described only in terms of $x_2, x_4$ (Refer to Eq. 4.16).
The above relations are then substituted into the following remaining $32 - 8 = 24$ equations:

\[
\begin{align*}
c_{24} &= \Theta(s_0, s_4, s_8, s_{11}) \oplus k_{80} \quad (4.17) \\
c_{25} &= \Theta(s_8, s_{11}, s_0, s_4) \oplus k_{82} \quad (4.18) \\
c_{26} &= \Theta(s_6, s_{10}, s_2, s_5) \oplus k_{88} \quad (4.19) \\
c_{27} &= \Theta(s_2, s_5, s_6, s_{10}) \oplus k_{90} \quad (4.20) \\
c_{28} &= \Theta(t_2, t_6, t_{10}, t_1) \oplus k_{100} \quad (4.21) \\
c_{29} &= \Theta(t_{10}, t_1, t_2, t_6) \oplus k_{102} \quad (4.22)
\end{align*}
\]

\[
\begin{align*}
s[c_0] &= \Pi(p_0, p_1, p_2, p_3) \oplus \Pi(k_0, k_1, k_2, k_3) \quad (4.29) \\
s[c_1] &= \Pi(p_8, p_9, p_6, p_7) \oplus \Pi(k_{10}, k_{11}, k_8, k_9) \quad (4.30) \\
s[c_2] &= \Pi(p_6, p_7, p_8, p_9) \oplus \Pi(k_8, k_9, k_{10}, k_{11}) \quad (4.31) \\
s[c_3] &= \Pi(p_2, p_3, p_0, p_1) \oplus \Pi(k_2, k_3, k_0, k_1) \quad (4.32) \\
s[c_4] &= \Pi(q_2, q_3, q_4, q_5) \oplus \Pi(k_{20}, k_{21}, k_{22}, k_{23}) \quad (4.33) \\
s[c_5] &= \Pi(q_{10}, q_{11}, q_8, q_9) \oplus \Pi(k_{30}, k_{31}, k_{28}, k_{29}) \quad (4.34) \\
s[c_6] &= \Pi(q_8, q_9, q_{10}, q_{11}) \oplus \Pi(k_{28}, k_{29}, k_{30}, k_{31}) \quad (4.35) \\
s[c_7] &= \Pi(q_4, q_5, q_2, q_3) \oplus \Pi(k_{22}, k_{23}, k_{20}, k_{21}) \quad (4.36) \\
s[c_8] &= \Pi(r_0, r_1, r_2, r_3) \oplus \Pi(k_{32}, k_{33}, k_{34}, k_{35}) \quad (4.37) \\
s[c_9] &= \Pi(r_8, r_9, r_6, r_7) \oplus \Pi(k_{42}, k_{43}, k_{40}, k_{41}) \quad (4.38) \\
s[c_{10}] &= \Pi(r_6, r_7, r_8, r_9) \oplus \Pi(k_{40}, k_{41}, k_{42}, k_{43}) \quad (4.39) \\
s[c_{11}] &= \Pi(r_2, r_3, r_0, r_1) \oplus \Pi(k_{34}, k_{35}, k_{32}, k_{33}) \quad (4.40)
\end{align*}
\]

Recall that the variables $s_j, t_j, u_j, p_j, q_j, r_j$ above are only temporary variables. For ease of notation, the exact values of these variables are given in Appendices C.1.1 and C.1.2.

After the substitutions, the system consists of $32 - 8 = 24$ equations in $12 - 8 = 4$ state and 108 subkey variables. The remaining state variables are $x_2, x_4, x_8$ and $x_{10}$. On closer inspection of the equations it can be seen that no more state variables can be eliminated using substitution, due to the nesting of the S-boxes. Recall that in order to construct this system, only 36 bytes of keystream generated consecutively under the same secret key are required.
4.2.2 Key Schedule Equations

The key schedule algorithm for LEX-AES is the same used in the AES, which is given in Section 2.4.1. The specific equations to compute the subkeys at round \( r \) are given as follows:

\[
\begin{align*}
    k_{16r} &= k_j \oplus s[k_{j+13}] \oplus RC^r \\
    k_{16r+1} &= k_{j+1} \oplus s[k_{j+14}] \\
    k_{16r+2} &= k_{j+2} \oplus s[k_{j+15}] \\
    k_{16r+3} &= k_{j+3} \oplus s[k_{j+12}] \\
    k_{16r+4} &= k_{j+4} \oplus k_{16r} \\
    k_{16r+5} &= k_{j+5} \oplus k_{16r+1} \\
    k_{16r+6} &= k_{j+6} \oplus k_{16r+2} \\
    k_{16r+7} &= k_{j+7} \oplus k_{16r+3} \\
    k_{16r+8} &= k_{j+8} \oplus k_{16r+4} \\
    k_{16r+9} &= k_{j+9} \oplus k_{16r+5} \\
    k_{16r+10} &= k_{j+10} \oplus k_{16r+6} \\
    k_{16r+11} &= k_{j+11} \oplus k_{16r+7} \\
    k_{16r+12} &= k_{j+12} \oplus k_{16r+8} \\
    k_{16r+13} &= k_{j+13} \oplus k_{16r+9} \\
    k_{16r+14} &= k_{j+14} \oplus k_{16r+10} \\
    k_{16r+15} &= k_{j+15} \oplus k_{16r+11}
\end{align*}
\]

where \( K^r = (k_{16r}, \ldots, k_{16r+15}) \), \( j = 16(r - 1) \), \( r = 1, 2, \ldots, 9 \) and \( RC^r \) is the round constant. Recall that round subkey bytes at iteration \( t \) are the same as the round subkey bytes at round \( r = t - \lfloor t/10 \rfloor \cdot 10 \), provided that the master key is unchanged. In essence, every byte of subkey \( K^r \) in round \( r \) depends on the values of at least one subkey byte \( K^{r-1} \) in round \( r - 1 \) where \( 1 \leq r < 10 \). Each of the first four subkey bytes is composed of the XOR of two different subkey bytes in the previous round. The remaining 12 subkey bytes are composed of the XOR of two subkey bytes: one from the current round and one from the previous round.

From the previous section, it is known that there are 108 subkey variables in the equations describing the keystream. The number of these subkey variables can be reduced by describing them in terms of only the 16 subkey variables in \( K^3 \). The key schedule equations are rearranged so that for instance, \( K^2 = \)
(\(k_{32}, \ldots, k_{47}\)) can be written in terms of \(K^3 = (k_{48}, \ldots, k_{63})\) only, as follows:

\[
\begin{align*}
    k_{32} &= k_{48} \oplus s[k_{61} \oplus k_{57}] \oplus RC^3 \\
    k_{33} &= k_{49} \oplus s[k_{62} \oplus k_{58}] \\
    k_{34} &= k_{50} \oplus s[k_{63} \oplus k_{59}] \\
    k_{35} &= k_{51} \oplus s[k_{60} \oplus k_{56}] \\
    k_{36} &= k_{52} \oplus k_{48} \\
    k_{37} &= k_{53} \oplus k_{49} \\
    k_{38} &= k_{54} \oplus k_{50} \\
    k_{39} &= k_{55} \oplus k_{51}
\end{align*}
\]

\[
\begin{align*}
    k_{40} &= k_{56} \oplus k_{52} \\
    k_{41} &= k_{57} \oplus k_{53} \\
    k_{42} &= k_{58} \oplus k_{54} \\
    k_{43} &= k_{59} \oplus k_{55} \\
    k_{44} &= k_{60} \oplus k_{56} \\
    k_{45} &= k_{61} \oplus k_{57} \\
    k_{46} &= k_{62} \oplus k_{58} \\
    k_{47} &= k_{63} \oplus k_{59}
\end{align*}
\]

In the general case, additional substitutions are required. For instance, the following describes \(K^4 = (k_{64}, \ldots, k_{79})\) in terms of \(K^3\) only:

\[
\begin{align*}
    k_{64} &= k_{48} \oplus s[k_{61}] \oplus RC^4 \\
    k_{65} &= k_{49} \oplus s[k_{62}] \\
    k_{66} &= k_{50} \oplus s[k_{63}] \\
    k_{67} &= k_{51} \oplus s[k_{60}] \\
    k_{68} &= k_{52} \oplus k_{48} \oplus s[k_{61}] \oplus RC^4 \\
    k_{69} &= k_{53} \oplus k_{49} \oplus s[k_{62}] \\
    k_{70} &= k_{54} \oplus k_{50} \oplus s[k_{63}] \\
    k_{71} &= k_{55} \oplus k_{51} \oplus s[k_{60}] \\
    k_{72} &= k_{56} \oplus k_{52} \oplus k_{48} \oplus s[k_{61}] \oplus RC^4 \\
    k_{73} &= k_{57} \oplus k_{53} \oplus k_{49} \oplus s[k_{62}] \\
    k_{74} &= k_{58} \oplus k_{54} \oplus k_{50} \oplus s[k_{63}] \\
    k_{75} &= k_{59} \oplus k_{55} \oplus k_{51} \oplus s[k_{60}] \\
    k_{76} &= k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus s[k_{61}] \oplus RC^4 \\
    k_{77} &= k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus s[k_{62}] \\
    k_{78} &= k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus s[k_{63}] \\
    k_{79} &= k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus s[k_{60}]
\end{align*}
\]

Similarly, the variables in \(K^0, K^1, K^5, K^6, K^7\) can be described in terms of \(K^3\). The full subkey substitution is given in Appendix C.1.3. These key schedule equations are then substituted in the keystream equations. Now, the system
consists of 24 equations in 4 state and $108 - 92 = 16$ subkey variables.

### 4.2.3 Additional Substitutions

By further inspection, it was discovered that three additional subkey variables can be eliminated. They are the variables $k_{59}$, $k_{51}$ and $k_{56}$. This is explained as follows. After the substitution, Equations 4.17 and 4.19 respectively, have the following forms:

\[
\begin{align*}
\Theta(v_0, v_1, v_2, k_{59} \oplus v_3) \oplus K^*_0 \oplus c_{24} &= 0 \quad (4.41) \\
\Theta(w_0, w_1, w_2, k_{51} \oplus w_3) \oplus K^*_1 \oplus c_{26} &= 0 \quad (4.42)
\end{align*}
\]

where the temporary variables $v_j, w_j$ and $K^*_j$ are as defined in Appendix C.1.4. A careful examination of Equation 4.41 reveals that it contains only a single subkey variable $k_{59}$. This variable can be removed by describing it as follows:

\[
k_{59} = s^{-1}[\Theta(v_0, v_1, v_2, 0) \oplus K^*_0 \oplus c_{24}] \oplus v_3.
\]

The above equation is then substituted into the remaining $24 - 1 = 23$ equations and the number of subkey variables in the system is reduced to 15.

The second subkey variable can be eliminated by using the fact that Equation 4.42 does not contain the subkey variable $k_{59}$. This means that Equation 4.42 is not affected by the previous substitution of $k_{59}$. It turns out that the subkey variable $k_{51}$ only appears once in Equation 4.42 and can be isolated and described in terms of the other variables as follows:

\[
k_{51} = s^{-1}[\Theta(w_0, w_1, w_2, 0) \oplus K^*_1 \oplus c_{26}] \oplus w_3.
\]

The above equation is then substituted into the remaining $23 - 1 = 22$ equations, and the number of subkey variables in the system is further reduced to 14.

The last subkey variable can be eliminated by observing the following equation, which is the updated form of Equation 4.18

\[
y_0 \oplus y_1 \oplus y_2 \oplus s[y_3 \oplus CBk_{56}] \oplus K^*_2 \oplus c_{25} = 0
\]

where the temporary variables $y_j$ and $K^*_2$ are defined in Appendix C.1.4. The above equation contains only a single subkey variable $k_{56}$ and can be isolated
and described in terms of other variables as follows:

\[ k_{56} = 4s^{-1}[y_0 \oplus y_1 \oplus y_2 \oplus K_2^* \oplus c_{25}] \oplus 4y_3 \]

The above equation is then substituted into the remaining \( 22 - 1 = 21 \) equations, and the number of subkey variables in the system is further reduced to 13. The system now consists of 21 equations in 17 variables (4 state and 13 subkey variables).

After the previous substitutions, no more state or subkey variables that can be eliminated, were found. This is because in the same equations, each remaining variable occurs both inside and outside of the S-box. For instance, in an equation of the form \( x \oplus s[x] \oplus \ldots = 0 \), where \( x \) is a variable, then it is not possible to isolate \( x \).

### 4.2.4 The Final System of Equations

The final equation system contains 21 equations in 17 variables, which represents a reduction in the number of equations from 32 to 21 and in the number of variables from 28 to 17. The equations are given in Appendix C.1.5. The equation system is also very straightforward to construct. Recall from Section 4.2.1 that the system requires only 36 bytes of known keystream, generated consecutively under the same secret key. In terms of required number of keystream bytes, this is very low and reasonably practical to obtain.

### 4.2.5 Solving the Equations

Note that for an efficient attack, the effort required to break the cipher must be less than performing an exhaustive search of the key space. One way of solving the system of equations is to guess the value of all variables and discard guesses for which the equations are inconsistent. There are 21 equations, which is four more than the number of variables in the system and thus, with very high probability, there is only one solution to the system. This solution must correspond to the correct key. If the number of variables is less than the number of key bytes (in our case, 16), then solving the system can be faster than exhaustive search.

Another possible way of solving the equations is to only guess selected variables. The reasoning for this is that the equations might be significantly simplified if a partial guess is made. The simplified equations are expected to be
4.2. Forming Equations to Describe LEX-AES

much easier to solve than the original equation system. In order for this to work, it must be possible to determine which subset of variables provides the greatest simplification for the equation system, and to be able to verify whether the partial guess is right or wrong, with high probability.

In the system of equations, there are many expressions that occur frequently. These frequent expressions are made up of the sum of some key or/and state variables. The common expressions are denoted as $W_j$, $Y_j$ and $Z_j$ in the final set of equations given in Appendix C.1.5. Some of these expressions contain as few as four variables; however, almost all of them contain the entire 17 variables. For example, consider the following equation, which is the last of the 21 equations given in Appendix C.1.5:

$$\hat{\Pi}(Z_{10}, Z_{11}, Z_8, Z_9) \oplus T_{60} \oplus s[c_{11}] = 0$$

where $Z_j$ are expressions, $\hat{\Pi}(z_0, z_1, z_2, z_3) = Es^{-1}[z_0] \oplus Bs^{-1}[z_1] \oplus Ds^{-1}[z_2] \oplus 9s^{-1}[z_3]$ and $T_{60}$ is composed of the sum of some state and key variables. The expressions $Z_9$ and $Z_{11}$ contain four variables while $Z_8$ and $Z_{10}$ contain the entire 17 variables. Even if the variables in $Z_9$ and $Z_{11}$ are guessed, there are still $17-4 = 13$ variables left in $Z_8$ and $Z_{10}$. In this case, the guesses do not simplify the equation. As a result, it was not possible to identify any subset of state or/and key variables that can be guessed to simplify any equation.

Figure 4.4: Keystream involved in blocks of 10 rounds.
4.2.6 Alternative Methods for Obtaining Equations

LEX-AES uses the same subkey every 10th round, provided that the master key is unchanged. This fact can be used to obtain additional equations by applying the previous technique of obtaining equations in blocks of 10 rounds. For each application, the number of equations and state variables will double but the number of round subkey variables remains unchanged. This is illustrated in Figure 4.4 where the dotted lines enclose the keystream and state bytes involved in the 10-round block. Let $m_t$ denote the number of 10-round blocks. If the technique is applied for $m_t$ 10-round blocks, then a system of $21m_t$ equations and $13 + 4m_t$ variables are obtained.

Another method of obtaining equations is to use only the constants contained in two iterations before and after the middle internal state. This requires only 20 consecutive bytes of the keystream. This system spans 4 rounds of the AES and uses the output keystream of 5 iterations. Initially, this system contains $4 \times 4 = 16$ equations in 12 state variables and $8 + 2 \times 16 + 4 = 44$ subkey variables. If the same middle iteration is used as before, the system starts at the output of $F^1_K$ and ends at the output of $F^5_K$. Refer again to Figure 4.4 for illustration. Assuming the same notations for variables and constants as before, Equations 4.9 to 4.16 can be used to substitute into the remaining $16 - 8 = 8$ equations. After this, the system is left with $12 - 8 = 4$ state variables. As explained in Section 4.2.2, the subkey variables can be substituted so that only 16 remain. The same technique can be used to eliminate three more subkey variables as outlined in Section 4.2.3. Assuming no more variables can be eliminated, the system is left with $8 - 3 = 5$ equations in $4 + 13 = 17$ variables which span 4 rounds of the AES. Similarly, the technique can be applied in blocks of 10 rounds which increases the number of equations and state variables.

Similarly, all constants that appear in three iterations before and after the middle internal state can be used. This requires 28 consecutive bytes of the keystream. The resulting equations span 6 rounds of the AES and uses the output keystream of 7 rounds. Initially, the system contains $6 \times 4 = 24$ equations in 12 state and $8 + 3 \times 16 + 4 = 60$ subkey variables. After the same elimination as before is performed, the system is left with $6 \times 4 - 11 = 13$ equations in $4 + 13 = 17$ variables. This technique can also be applied to blocks of 10 rounds which increases the number of equations and state variables.

The final form for the sets of equations that span 4 and 6 rounds of the LEX-
AES are simpler than the previous system that spans 8 rounds. However, these systems are underdefined if the systems comprise only one 10-round block. If the systems comprise more than one 10-round block, then the amount of keystream required is greater, but still considerably low.

4.3 Forming Equations in Small Scale Variants of LEX-BES

The system of equations arising from LEX-BES is very similar to those generated from the BES technique. The difference is that, due to the keystream leaks, there are more known values in the LEX-BES equations compared to BES. The system of equations in LEX-BES contains linear and nonlinear relationships in the components of the AES round function. This section presents the results of experiments on algebraic cryptanalysis of small scale variants of LEX-BES. Background information regarding BES, LEX-BES and small scale variants of LEX-BES are explained in the following subsections before the results are presented.

4.3.1 BES

The Big Encryption System (BES) is a block cipher that accepts a 128-byte plaintext block $x_0$ and a master key $k$ with allowable sizes of 128, 192 and 256 bytes. However, this thesis focuses only on the BES with 128-byte master key. The master key is used as input to the key scheduling algorithm to produce a set of eleven 128-byte round subkeys. The ciphertext block $x_{11}$ is produced after the round function is applied 10 times.

The BES cipher is an extension of the AES and thus has the same number of rounds and similar round function transformations. The AES can be seen as the BES but with much smaller plaintext and key spaces. The AES operates in both $\mathbb{F}_2$ and $\mathbb{F}_{2^8}$ finite fields. In contrast, the BES operates solely in $\mathbb{F}_{2^8}$ and this is said to facilitate the algebraic analysis of the AES. Furthermore, the only nonlinear component of the AES is described simply using the inversion operation.

In the AES, the 16-byte state block can be regarded as a column vector of length 16. A byte is considered as an element in the binary field defined by the
following irreducible polynomial: \( x^8 + x^4 + x^3 + x + 1 \). Let \( \theta \) be a root of this polynomial and let the field \( F \) be defined as

\[
F = \mathbb{F}_{2^8} = \frac{\mathbb{F}_2[x]}{x^8 + x^4 + x^3 + x + 1} = \mathbb{F}_2(\theta).
\]

Therefore, each byte can be represented as a polynomial in \( \theta \). In this chapter, the term \( \theta^7 \) corresponds to the leftmost bit of a hexadecimal representation of a byte. For instance, the polynomial \( \theta^6 + \theta^3 \) corresponds to the hexadecimal value \( 48 \).

Let \( A \) denote the state space of the AES which is the vector space \( \mathbb{F}^{16} \) and let \( B \) denote the state space of the BES which is the vector space \( \mathbb{F}^{128} \). A state vector of the AES in round \( r \) is an element \( a_r \in A \) defined as the column vector \( a_r = (a_r(0), a_r(1), \ldots, a_r(15)) \) of length 16. Similarly, a state vector of the BES in round \( r \) is an element \( b_r \in B \) defined as the column vector \( b_r = (b_r(0,0), \ldots, b_r(0,7), b_r(1,0), \ldots, b_r(15,7)) \) of length \( 16 \cdot 8 = 128 \).

In order to embed any element of the AES state space \( A \) into the BES state space \( B \), a vector conjugate mapping \( \phi \) from \( \mathbb{F} \) to a subset of \( \mathbb{F}^8 \) is used. For any element \( a_r(i) \in \mathbb{F} \), the mapping \( \phi \) is defined as follows:

\[
\phi(a_r(i)) = (a_{r(i)}^2, a_{r(i)}^1, a_{r(i)}^2, a_{r(i)}^2, a_{r(i)}^4, a_{r(i)}^2, a_{r(i)}^6, a_{r(i)}^2)
\]

that is \( b_{r(i,j)} = a_{r(i)}^j \). The vector \( (b_{r(i,0)}, \ldots, b_{r(i,7)}) \) is called the vector conjugate of \( a_{r(i)} \). Similarly, any vector \( a_r = (a_r(0), a_r(1), \ldots, a_r(15)) \in \mathbb{F}^{16} \) is mapped to a subset of \( \mathbb{F}^{128} \) as follows:

\[
\phi(a_r) = (\phi(a_r(0)), \phi(a_r(1)), \ldots, \phi(a_r(15)))
\]

\[
= (b_{r(0,0)}, \ldots, b_{r(0,7)}, b_{r(1,0)}, \ldots, b_{r(15,7)}).
\]

The embedded image of the AES state space \( A \) in the BES state space \( B \) denoted by \( B_A \), is defined by

\[
B_A = \phi(A) \subset B.
\]
4.3. Forming Equations in Small Scale Variants of LEX-BES

Encryption Algorithm

The 128-byte plaintext block \( x_0 = (x_{0(0,0)}, \ldots, x_{0(0,7)}, x_{0(1,0)}, \ldots, x_{0(15,7)}) \) is defined as a column vector of length 128. Let \( k_r = (k_{r(0,0)}, \ldots, k_{r(0,7)}, k_{r(1,0)}, \ldots, k_{r(15,7)}) \) denote the column vector of the 128-byte subkey in round \( r \). Let \( L_S, L_W \), and \( L_M \) represent \( 128 \times 128 \) matrices over \( \mathbb{F}_2 \), which correspond to the BES representations of the \( \mathbb{F}_2 \)-matrix component of the S-box, the ShiftRows and MixColumns transformations of the AES, respectively. The matrix entries are outlined in the following text. The \( R \) rounds of the BES can be expressed by the following equations

\[
\begin{align*}
    x_1 &= x_0 + k_0, \\
    x_{r+1} &= L_M \cdot L_W \cdot L_S \cdot x_r^{-1} + k_r, \quad r = 1, 2, \ldots, R - 1, \\
    x_{R+1} &= L_W \cdot L_S \cdot x_R + k_R,
\end{align*}
\]

where a zero inversion is defined as \( 0^{-1} = 0 \).

The \( 128 \times 128 \) matrix \( L_S \) over \( \mathbb{F}_2 \) corresponds to the BES representation of the \( 8 \times 8 \mathbb{F}_2 \)-matrix component of the AES S-box. The matrix \( L_S = \text{diag}(L_0, \ldots, L_S) \) is a block diagonal matrix with 16 identical blocks \( L_S \), where \( L_S \) is defined as follows:

\[
L_S = \begin{pmatrix}
    \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \lambda_7 \\
    (\lambda_7)^2 & (\lambda_0)^2 & (\lambda_1)^2 & (\lambda_2)^2 & (\lambda_3)^2 & (\lambda_4)^2 & (\lambda_5)^2 & (\lambda_6)^2 \\
    (\lambda_6)^2 & (\lambda_7)^2 & (\lambda_0)^2 & (\lambda_1)^2 & (\lambda_2)^2 & (\lambda_3)^2 & (\lambda_4)^2 & (\lambda_5)^2 \\
    (\lambda_5)^2 & (\lambda_6)^2 & (\lambda_7)^2 & (\lambda_0)^2 & (\lambda_1)^2 & (\lambda_2)^2 & (\lambda_3)^2 & (\lambda_4)^2 \\
    (\lambda_4)^2 & (\lambda_5)^2 & (\lambda_6)^2 & (\lambda_7)^2 & (\lambda_0)^2 & (\lambda_1)^2 & (\lambda_2)^2 & (\lambda_3)^2 \\
    (\lambda_3)^2 & (\lambda_4)^2 & (\lambda_5)^2 & (\lambda_6)^2 & (\lambda_7)^2 & (\lambda_0)^2 & (\lambda_1)^2 & (\lambda_2)^2 \\
    (\lambda_2)^2 & (\lambda_3)^2 & (\lambda_4)^2 & (\lambda_5)^2 & (\lambda_6)^2 & (\lambda_7)^2 & (\lambda_0)^2 & (\lambda_1)^2 \\
    (\lambda_1)^2 & (\lambda_2)^2 & (\lambda_3)^2 & (\lambda_4)^2 & (\lambda_5)^2 & (\lambda_6)^2 & (\lambda_7)^2 & (\lambda_0)^2 
\end{pmatrix}
\]

and \( (\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7) = (5, 9, F9, 25, F4, B5, B9, 8F) \).

The \( 128 \times 128 \) matrix \( L_W \) over \( \mathbb{F}_2 \) corresponds to the BES representation of the linear transformation \( L_0 \) (or ShiftRows) of the AES. The matrix \( L_W \) is a
Chapter 4. Algebraic Analysis of LEX-AES

A block matrix defined as follows:

\[
L_W = \begin{pmatrix}
I_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I_8 & 0 & 0 & 0 \\
0 & I_8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I_8 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

where \(I_8\) denotes the \(8 \times 8\) identity matrix.

The \(128 \times 128\) matrix \(L_M\) over \(\mathbb{F}\) corresponds to the BES representation of the linear transformation \(L_1\) (or \texttt{MixColumns}) of the AES. The matrix \(L_M\) is composed of the following eight versions of the \texttt{MixColumns} \(4 \times 4\) matrix over \(\mathbb{F}\):

\[
\hat{L}_W^{(j)} = \begin{pmatrix}
\gamma_0 2^j & \gamma_1 2^j & \gamma_2 2^j & \gamma_3 2^j \\
\gamma_3 2^j & \gamma_0 2^j & \gamma_1 2^j & \gamma_2 2^j \\
\gamma_2 2^j & \gamma_3 2^j & \gamma_0 2^j & \gamma_1 2^j \\
\gamma_1 2^j & \gamma_2 2^j & \gamma_3 2^j & \gamma_0 2^j
\end{pmatrix}
\]

for \(j = 0, 1, \ldots, 7\) where \((\gamma_0, \gamma_1, \gamma_2, \gamma_3) = (2, 3, 1, 1)\). If \(a_{i,j} \in \mathbb{F}\) denote the elements of the matrix \(L_M\), then Algorithm 4.1 can be used to compute their values.

**Key Scheduling Algorithm**

Similar to the AES, the key scheduling algorithm accepts 128-, 192- and 256-byte master keys. The master key \(k = (k_{(0,0)}, \ldots, k_{(0,7)}, k_{(1,0)}, \ldots, k_{(15,7)})\) is formed from the concatenation of 128 bytes \(k_{(i,j)}\). The description of the key scheduling algorithm for the BES is similar to the AES described in Section 2.4.1. The operation on bytes in the AES is performed similarly to the corresponding vector conjugates in the BES.
begin
Set the value of all elements in the matrix $L_M = [a_{i,j}]_{128 \times 128}$ to zero;
for $n_c = 0$ to 3 do
  for $n_r = 0$ to 3 do
    for $i = 0$ to 7 do
      for $j = 0$ to 3 do
        $a_{32n_c+8n_r+i,32n_c+8j+i} = (\gamma_{j-n_r})^{2^i};$
      end
    end
  end
end
end

Algorithm 4.1: Computing the elements of the matrix $L_M$.

### 4.3.2 LEX-BES

LEX-BES is very similar to LEX-AES, which is described in Section 2.4.7. Here, only the parts which are different are mentioned.

#### Initialization

The eleven round subkeys are denoted by $k_0, k_1, \ldots, k_{10}$. The internal state is initialized by encrypting a 128-byte IV with $k$ using the full 10-round BES. The output of the initialization phase is denoted by $x_0$.

#### Keystream Generation

The 128-byte internal state $x_t = (x_{t(0,0)}, \ldots, x_{t(0,7)}, x_{t(1,0)}, \ldots, x_{t(15,7)})$ is updated in iteration $t$ using the round function of LEX-BES. The $T$ iterations of the keystream generator of LEX-BES can be described as follows:

$$x_{t+1} = L_M \cdot L_W \cdot L_S \cdot x_t^{-1} + k_{t \mod 10}, \quad t = 0, 1, \ldots, T - 1.$$  

In every iteration, after the state is updated, $4 \times 8 = 32$ bytes of $x_t$ are extracted directly by the output function $f$ to form the keystream. The extracted bytes are

$$f(x_t) = \begin{cases} (x_{t(0)}, x_{t(2)}, x_{t(8)}, x_{t(10)}), & \text{if } t \text{ is odd} \\ (x_{t(4)}, x_{t(6)}, x_{t(12)}, x_{t(14)}), & \text{if } t \text{ is even} \end{cases}$$

where $x_{t(i)} = (x_{t(i,0)}, x_{t(i,1)}, \ldots, x_{t(i,7)})$. This implies that there are redundancies in the keystream because for every byte, there are seven other bytes that are
derived from this one byte, that is \( x_{t(i,j)} = x_{t(i,j-1)}^2 \) for \( j = 1, 2, \ldots, 7 \).

### Equation System for LEX-BES

Let the vector \( x_t \) denote the inverse of the vector \( w_t \) at iteration \( t \), that is, \( x_t = w_t^{-1} \). Let \( x_{t(i,j)}, w_{t(i,j)} \) and \( k_{t(i,j)} \) denote the \((8i + j)\)-th component of the vectors \( x_t, w_t \) and \( k_t \), respectively. The following system of equations describes \( T \) iterations of LEX-BES:

\[
\begin{align*}
0 &= x_{t(i,j)}w_{t(i,j)} + 1 & t &= 0, \ldots, T \quad (4.43) \\
0 &= w_{t(i,j)} + k_{t \mod 10(i,j)} + \sum \alpha_{(i,j')}x_{t-1(k',m')} & t &= 1, \ldots, T \quad (4.44) \\
0 &= x_{t(i,j)}^2 + x_{t(i,j+1)} & t &= 1, \ldots, T \quad (4.45) \\
0 &= w_{t(i,j)}^2 + w_{t(i,j+1)} & t &= 1, \ldots, T \quad (4.46)
\end{align*}
\]

where the \( w_{0,(i,j)} \) are the input vector components for the first iteration of LEX-BES (the output of the initialization phase), and the \( \alpha_{(i,j')} \) represent the elements of the matrix \( L_M \cdot L_W \cdot L_S \). The above system of equations is assumed to have no zero inversion. In other words, the vector components \( x_{t(i,j)}, w_{t(i,j)} \) and \( k_{t(i,j)} \) are assumed to be nonzero.

Similar to the equations described in Section 4.2, it is assumed that the keystream bytes are known. In LEX-BES, the round subkey used in the first iteration is \( k_1 \) and the value of \( w_0 \) is unknown. For simplicity in writing the equations, the iteration will be started where the round subkey used is \( k_0 \). Therefore, the number of encryption equations for 10 iterations is \( 10 \cdot 8(16 + 3 \cdot 12) = 4160 \) in \( 8 \cdot 12(10 \cdot 2 + 1) = 2016 \) state and \( 10 \cdot 8 \cdot 16 = 1280 \) key variables. The system includes \( 10 \cdot 8 \cdot 3 \cdot 12 = 2880 \) quadratic and 1280 linear equations. Since only 10 round subkeys are used in LEX-BES, the key schedule can be written using 3008 equations in 1280 basic and 288 auxiliary key equations.

More equations can be generated by repeating the system by going across 10-round blocks. If consecutive 10-round blocks are used, then each repetition will add \( 416 \times 10 = 4160 \) equations and \( 192 \times 10 = 1920 \) variables. The number of key schedule equations and round subkey variables remain unchanged because the same subkeys are used in every 10-th round. If more than one 10-round block is used, then the blocks are assumed consecutive.

In comparison, the BES encryption can be written as a system of 5248 equations over \( GF(2^8) \), of which 3840 are quadratic and the remaining 1408 are linear.
Table 4.1: Comparison between BES\((n, r, c, e)\) and LB\((n, r, c, e)\) in terms of the number of equations and variables.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BES((n_R, n_w, n_c, n_b))</td>
<td>((4n + 1)n_wn_cn_b)</td>
<td>(2n_Rn_wn_cn_b)</td>
<td>((n_R + 1)n_wn_cn_b)</td>
<td>((n_c + 2n_Rn_c + 2n_R)n_wn_b)</td>
<td>(n_Rn_wn_b)</td>
</tr>
<tr>
<td>LB((n_R, n_w, n_c, n_b))</td>
<td>(13n_wn_cn_b/4)</td>
<td>(9n_wn_cn_b/4 + 3n_wn_cn_b(t - 1)/2)</td>
<td>(n_Rn_wn_cn_b)</td>
<td>(n_wn_cn_b + (n_R - 1)(2n_wn_cn_b + 2n_wn_b))</td>
<td>((n_R - 1)n_wn_b)</td>
</tr>
</tbody>
</table>

There are 2560 state variables and 1408 key variables in this system. In addition, the key schedule can be described using a similar system comprising 3328 equations in 1408 basic and 320 auxiliary key variables. The number of equations and variables arising from small scale LEX-BES and BES is summarized in Table 4.1. As noted earlier, the parameter \(t\) denotes the number of LEX-BES iterations.

### 4.3.3 Small Scale LEX-BES

This section briefly describes the small scale LEX-BES which is based on the small scale AES proposed by Cid, Murphy and Robshaw \[46\]. Several small scale variants of LEX-BES are used in the experiments presented in Section 4.3.4.

The small scale LEX-BES parameters are given by \(n_R, n_w, n_c, n_b\). The parameter \(n_R\) denotes the number of BES rounds used in LEX-BES. The parameters \(n_w\) and \(n_c\) denote the number of rows and columns, respectively in the AES state matrix\[3\]. The parameter \(n_b\) denotes both the size of each element in the AES state matrix and the number of elements in a vector conjugate of LEX-BES. Let LB\((n_R, n_w, n_c, n_b)\) denote a specific instance of a small scale LEX-BES.

### Equation System for Small Scale LEX-BES

The system of equations for small scale LEX-BES is very similar to those given for the full scale LEX-BES. In fact, the equation system for the full scale LEX-BES corresponds to the LB\((10, 4, 4, 8)\) variant. The number of equations and variables for the small scale variant LB\((10, 2, 2, 4)\) are shown in Table 4.2. For instance, the entries for \(t = 11\) denote the total number of equations and variables that exist in the first 11 iterations. In the first 10 iterations, the number of variables

---

\[3\] The AES state matrix has four rows and four columns, as depicted in Figure 2.12.
Table 4.2: Number of equations and variables for LB(10, 2, 2, 4)

<table>
<thead>
<tr>
<th>Iteration (t)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equations</td>
<td>68</td>
<td>168</td>
<td>268</td>
<td>368</td>
<td>468</td>
<td>568</td>
<td>668</td>
<td>768</td>
<td>868</td>
<td>968</td>
</tr>
<tr>
<td>Variables</td>
<td>52</td>
<td>100</td>
<td>148</td>
<td>196</td>
<td>244</td>
<td>292</td>
<td>340</td>
<td>388</td>
<td>436</td>
<td>484</td>
</tr>
</tbody>
</table>

Table 4.3: Number of equations and variables for small scale BES defined over $GF(2^4)$

<table>
<thead>
<tr>
<th>Rounds ($n_R$)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equations</td>
<td>144</td>
<td>256</td>
<td>368</td>
<td>480</td>
<td>592</td>
<td>704</td>
<td>816</td>
<td>928</td>
<td>1040</td>
<td>1152</td>
</tr>
<tr>
<td>Variables</td>
<td>72</td>
<td>128</td>
<td>184</td>
<td>240</td>
<td>296</td>
<td>352</td>
<td>408</td>
<td>464</td>
<td>520</td>
<td>576</td>
</tr>
</tbody>
</table>

in each iteration is increased by 48 from the previous iteration. However, the number of variables increased only by 24 when $t > 10$ since the number of round subkey variables remains unchanged.

Compared to LEX-BES (and its small scale variants), the total number of equations arising from BES is always twice as many as the number of variables. An example is shown in Table 4.3 for a small scale BES defined over $GF(2^4)$ where both the number rows and columns of its state matrix is four. In LEX-BES, for $t > 10$, as the number of iterations increases, the number of equations will also gradually increase more than the increase in variables.

### 4.3.4 Experimental Results

This section presents experimental results which relate to the solution of the equation system describing small scale LEX-BES. The experiments show the time required to find the solution for the equations using Gröbner basis computations. See Section 2.3.9 for a brief explanation of Gröbner basis. The computations were performed using the MAGMA 2.14-11 computational algebra package. The experiments were implemented on a 64-bit 1.6 GHz Itanium 2 processor supercomputer with 198 GB of RAM, running on SUSE Linux operating system. This supercomputer is provided by the High Performance Computing (HPC) & Research Support group of Queensland University of Technology (QUT), Australia.

The experiments were conducted as follows. Firstly, the system of polynomial equations consisting of the small scale LEX-BES keystream generation and key scheduling algorithm were written. Then, the corresponding polynomial
ideal generated from the polynomials in the system was computed using Magma. Lastly, the Gröbner basis of the related polynomial ideal was computed. If a unique solution was found, Gröbner basis immediately gives solution to the original system. Since the equations of the form \( xw = 1 \) are invalid when \( x = w = 0 \), the experiments were configured so that no zero-inversion occurs in the system.

Table 4.4 shows the results of finding solutions for the equation system arising from LB(1, 2, 2, 4) using Gröbner basis computations. This variant uses the same subkey in every iteration of the update function. The time to compute the Gröbner basis for the equations in a single iteration is very fast, i.e., about 2 seconds, and the memory requirement is very small. However, by constructing the equations over 2 iterations, MAGMA failed to compute the Gröbner basis solution after 5 days of running time. Since the calculation is not done in a reasonable time, it is considered to be non-feasible.

Table 4.5 shows the results of computing the Gröbner basis solution for the equation system arising from LB(10, 2, 2, 4). The number of 10-round blocks denotes how many times the equations are repeated for the next consecutive 10-round block. For instance, if no 10-round block is used, then the equation system comprises only one iteration, say iteration \( i \). If one 10-round block is used, then the equation system comprises the equations in iteration \( i \) and iteration \( i + 10 \). In the first row of Table 4.5, the system contains 68 equations in 52 variables. The 68 equations comprise of 12 equations each of the form given in Equation 4.43, Equation 4.45, and Equation 4.46 (3 \( \times \) 12 = 36 in total); 16 equations of the form given in Equation 4.44; and 16 equations of the form given in Equations 4.44 but for the key scheduling equations. The 52 variables comprise of 3 \( \times \) 12 = 36 internal state variables (the variables \( x_{0,(i,j)} \), \( w_{0,(i,j)} \) and \( w_{1,(i,j)} \)) and 16 subkey variables (\( k_{1,(i,j)} \)). The addition of one 10-round block adds 52 new equations (the encryption equations only) and 36 new variables (the internal state).

The experiments failed to compute the Gröbner basis solution in reasonable time after 2 iterations, even for a small number of variables (less than 100). The experiments performed here were similar to those by Cid, Murphy and Robshaw.

Table 4.4: Time (in seconds) and memory (in MB) required to compute Gröbner basis for the equation system arising from LB(1, 2, 2, 4)

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Equations</th>
<th>Variables</th>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68</td>
<td>52</td>
<td>0.03</td>
<td>7.5</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>76</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 4.5: Time (in seconds) and memory (in MB) required to compute Gröbner basis for the equation system arising from LB(10, 2, 2, 4)

<table>
<thead>
<tr>
<th>Number of 10-round blocks</th>
<th>Equations</th>
<th>Variables</th>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>68</td>
<td>52</td>
<td>1.8</td>
<td>7.40</td>
</tr>
<tr>
<td>1</td>
<td>120</td>
<td>88</td>
<td>4,497.6</td>
<td>727.89</td>
</tr>
<tr>
<td>3</td>
<td>172</td>
<td>124</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

[46]. In their results, they failed to compute the Gröbner basis for a system of equations arising from AES(2, 2, 2, 4) and onwards. The results in this chapter seem to complement their work. Therefore, since it was not possible to get feasible results on small scale variants of LEX-BES, it is conjectured that the same experiments would most likely to fail on the full scale of LEX-BES, which contains a system of thousands of equations and variables.

4.4 Discussion

Note that even though the work presented in this chapter does not provide a key recovery attack and the work of Dunkelman and Keller [68, 69] does, it is still important for two reasons. First, as always with algebraic attacks, very little known keystream is needed. This makes an attack by solving an equation system needing 36 bytes of known keystream more threatening in a real-world situation than attacks needing almost 85 billion ($2^{263}$) bytes of known keystream. Recall from Section 2.4.7 that for a more secure variant of LEX-AES, the amount of keystream produced by one key and IV pair should not exceed 2000 bytes. The approach in this chapter, therefore, can be applied to this variant while previous attacks cannot. Second, the constructed equation system is almost sufficient for an efficient attack. While guessing 16 bytes is the limit for an efficient attack, only 17 bytes need to be guessed in order to solve the system of equations presented in this chapter. Traditionally, attacks on block ciphers have been classified as to how many rounds a particular attack is able to break, out of the full number of rounds. The small difference between 17 and 16 leads one to think that the attack presented here has the same strength as an attack on a block cipher that falls only one or two rounds short of breaking the full cipher.

No clever tricks were used in constructing our equation system in this chapter. It is a tedious but straightforward job to construct the system and then start eliminating variables. In particular, no use was made of the algebraic properties
of the S-box. As known from literature [62], the S-box can be replaced with the following polynomial equation:

\[ s[x] = 5x^{254} \oplus 9x^{253} \oplus F9x^{251} \oplus 25x^{247} \oplus F4x^{239} \oplus B5x^{223} \oplus B9x^{191} \oplus 8Fx^{127} \oplus 63 \]

The inverse S-box polynomial equation, however, is more dense than the above equation. It contains 247 terms and is given in Appendix A of the paper by Buchmann et al. [39]. The complicated structure of the inverse equations makes solving the equations very hard. The degree of the resulting equation system is also expected to be very high. This leaves room for further research. It is possible that by exploiting the properties of the S-box, one can improve on the results obtained in this chapter.

4.5 Summary and Conclusion

In this chapter, it is shown that the security of LEX-AES against algebraic attacks relies on the difficulty of solving a very small system of equations. An equation system can be constructed which spans eight rounds of the AES and contains only 21 equations and 17 variables (4 state and 13 subkey variables). Initially, the system involves 32 equations and 120 variables (12 state and 108 subkey variables). However, in this work, the elimination of 11 equations and 103 variables (8 state and 95 subkey variables) was achieved. This is a massive reduction in terms of the number of variables and is very close to the upper limit for an efficient attack, i.e. 16 variables.

The system of equations of LEX-AES in the context of the BES embedding was also investigated. This system gives a set of 4160 equations in 3296 variables. If the key scheduling algorithm is taken into account, this will add another 3008 equations in 1568 variables. For small scale experiments, it is possible to compute the Gröbner basis solution for a very small number of equations and variables. However, even though the degree of the equations are not very high (quadratic), the same computation fails for a system of 120 equations and 76 variables over $GF(2^4)$. Therefore, applying the same experiments to the full LEX-BES which contains thousands of equations and variables over $GF(2^8)$ does not seem to be a feasible option.
Note that LEX is intended as a generic method for constructing a stream cipher from a block cipher. In this chapter, a specific instance of LEX which uses the AES has been explored in terms of building a small system of equations. The AES is known to be a very strong cipher. Yet it has been shown in this chapter that the resulting equation system is very close to the threshold for key recovery. If other, possibly weaker block ciphers are used in this manner, then the security of the resulting stream cipher is surely questionable. This remains an area of further investigation, and it is clear that the results in this chapter indicate that a thorough investigation of possible algebraic attacks is required when using the LEX design with a different block cipher than the AES.
Chapter 5

Diffusion in the Linear Transformations of SPN Block Ciphers

The linear diffusion transformation of a Substitution-Permutation Network (SPN) block cipher is an important component of the round function. The transformation provides diffusion [162] by linearly mixing bits of the fixed-size input block to produce the corresponding output block of the same size. Existing techniques of measuring diffusion include the avalanche effect [75] and the strict avalanche criterion (SAC) [175]. These measures quantify the effects of a one-bit input change to changes in the output bits. Another measure is the completeness property, which deals with the dependency of each of the output bits on the input bits [94]. A more recent measure of diffusion specifically for SPN block ciphers is the branch number [62]. The branch number denotes the minimum number of active S-boxes for any two consecutive rounds. The number can be used to estimate the success of differential and linear attacks on a particular block cipher.

To date, it is believed that there is no existing technique that measures how well the linear transformation effectively changes the value of the input blocks to produce the output blocks. In this chapter, a new and simple method of measuring the diffusion provided by the linear transformation of SPN block ciphers is proposed. The measure is based on the number of fixed points and provides
an indication of how well the linear transformation effectively changes the value of the input block when producing the output block. An input block is a fixed point of a transformation if the input block equals its output block. Clearly, in this context, there is no diffusion at the fixed points since the input blocks are left unchanged by the linear transformation. Note that the proposed method is not intended to replace the branch number or other existing techniques. Rather, it is intended as an additional measure, to complement them.

The existence of fixed points in the round function of block ciphers has been used as the basis for a number of attacks. In particular, the block ciphers DES [51], SAFER K [110], Blowfish [97], GOST, DEAL [96] and KeeLoq [54] were previously found to be vulnerable to attacks based on the existence of fixed points. These attacks use fixed points that exist across one or more rounds. This work, however, highlights fixed points in the linear diffusion transformation of SPN block ciphers.

In this chapter, the new proposed diffusion measure is applied to the linear diffusion transformations of several SPN block ciphers: the Advanced Encryption Standard (AES) [62], ARIA [116, 146], Present [37], and Serpent [4]. It is shown that the linear diffusion transformations of all ciphers except Serpent have more fixed points than the expected number for a random linear transformation. Techniques to exploit the existence of these fixed points in an attack are also proposed.

This chapter is organized as follows. Section 5.1 describes some background concepts regarding permutations and fixed points. The new proposed method of measuring diffusion is explained in Section 5.2. Section 5.3 presents the application of the method to a number of SPN block ciphers. Section 5.4 contains the discussion of the relationship between fixed points and existing design criteria for linear diffusion transformation. The cryptographic significance of fixed points in the linear diffusion transformation is given in Section 5.5. Section 5.6 highlights some related work. Conclusions regarding the existence of fixed points are given in Section 5.7.

5.1 Preliminaries

This section explains some preliminary concepts regarding permutations and fixed points, to provide the context for the subsequent sections.
Table 5.1: Theoretical probabilities that random permutations of \( n \) elements have \( c \) fixed points.

<table>
<thead>
<tr>
<th>( n )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.33333</td>
<td>0.5</td>
<td>0</td>
<td>0.16667</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.375</td>
<td>0.33333</td>
<td>0.25</td>
<td>0</td>
<td>0.04167</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.36667</td>
<td>0.375</td>
<td>0.16667</td>
<td>0.08333</td>
<td>0</td>
<td>0.00833</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.36806</td>
<td>0.36667</td>
<td>0.18750</td>
<td>0.05556</td>
<td>0.02083</td>
<td>0</td>
<td>0.00139</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.36786</td>
<td>0.36806</td>
<td>0.18333</td>
<td>0.0625</td>
<td>0.01389</td>
<td>0.00417</td>
<td>0</td>
<td>0.0002</td>
</tr>
<tr>
<td>8</td>
<td>0.36788</td>
<td>0.36786</td>
<td>0.18403</td>
<td>0.06111</td>
<td>0.01563</td>
<td>0.00278</td>
<td>0.00069</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0.36788</td>
<td>0.36788</td>
<td>0.18394</td>
<td>0.06134</td>
<td>0.01528</td>
<td>0.00313</td>
<td>0.00046</td>
<td>0.0001</td>
</tr>
<tr>
<td>10</td>
<td>0.36788</td>
<td>0.36788</td>
<td>0.18394</td>
<td>0.06131</td>
<td>0.01534</td>
<td>0.00306</td>
<td>0.00052</td>
<td>0.00007</td>
</tr>
<tr>
<td>11</td>
<td>0.36788</td>
<td>0.36788</td>
<td>0.18394</td>
<td>0.06131</td>
<td>0.01533</td>
<td>0.00307</td>
<td>0.00051</td>
<td>0.00007</td>
</tr>
<tr>
<td>12</td>
<td>0.36788</td>
<td>0.36788</td>
<td>0.18394</td>
<td>0.06131</td>
<td>0.01533</td>
<td>0.00307</td>
<td>0.00051</td>
<td>0.00007</td>
</tr>
</tbody>
</table>

### 5.1.1 Fixed Points in Random Permutations

Consider a set of \( n \) elements denoted \( S = \{0, 1, \ldots, n - 1\} \). A permutation of \( S \) is a bijective map of the set \( S \) onto itself. An element of \( S \) is a fixed point of this map if it is invariant under this mapping. There are \( n! \) possible permutations of the set \( S \). The probability that a given permutation of the set \( S \) has \( c \) fixed points is [156, Chap. 3]:

\[
p_{n,c} = \frac{1}{n!} \cdot \binom{n}{c} \cdot (n - c)! \cdot \sum_{k=0}^{n-c} \frac{(-1)^k}{k!} \approx \frac{1}{c!e}.
\]

The values of \( p_{n,c} \) for \( n = 1, 2, \ldots, 12 \) and \( c = 0, 1, \ldots, 7 \) are given in Table 5.1.

For \( c = 0 \) and \( c = 1 \), as the number \( n \) of elements tends to infinity, both \( p_{n,0} \) and \( p_{n,1} \) approach \( e^{-1} = 0.3679 \). Therefore, the probability that a random permutation has more than one fixed point for large \( n \) is approximately \( 1 - 2(0.3679) = 0.2642 \). Note that for \( n > 9 \) the values of \( p_{n,0} \) and \( p_{n,1} \) are already very close to the limiting value. Note also that the expected number of fixed points in a random permutation is one [82, Chap. 6].

### 5.1.2 Fixed Points in Linear Transformations

Some permutations exist where the bijective map is a linear transformation which can be represented by nonsingular matrices. The following notations are used
in this chapter. Consider $b$-bit values as elements in the field $\mathbb{F}_{2^b}$ and let $S$ denote the set of all vectors over $\mathbb{F}_{2^b}$ of length $m$, i.e. $\mathbb{F}_{2^b}^m$. Define $A = [a_{i,j}]_{m \times m}$ as a nonsingular matrix with elements in the field $\mathbb{F}_{2^q}$ that represents a linear transformation $L$ on the elements of $S$. The field $\mathbb{F}_{2^q}$ is a subset of $\mathbb{F}_{2^b}$ and for the ciphers analyzed, only the cases where $q = 1$ or $q = 8$ and $b = 1$ or $b = 8$ are considered. The transformation $L$ maps an element $Z = (Z_0, Z_1, \ldots, Z_{m-1})^T \in S$ to an element $X = (X_0, X_1, \ldots, X_{m-1})^T \in S$ by $X = AZ$ as follows:

$$
\begin{pmatrix}
X_0 \\ X_1 \\ \vdots \\ X_{m-1}
\end{pmatrix} =
\begin{pmatrix}
a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,m-1}
\end{pmatrix}
\begin{pmatrix}
Z_0 \\ Z_1 \\ \vdots \\ Z_{m-1}
\end{pmatrix}
$$

(5.1)

where $a_{i,j} \in \mathbb{F}_{2^q}$ and $Z, X \in \mathbb{F}_{2^b}^m$. In other words, the equation above represents a mapping from $mb$ bits to $mb$ bits.

Let $I$ denote the $m \times m$ identity matrix. The set of all fixed points for a linear transformation $L$ that can be represented by a nonsingular matrix $A$ can be obtained by solving the following equation

$$(A - I)Z = 0$$

(5.2)

where $0$ is the all-zero vector of length $m$. The number of fixed points for this transformation is given by

$$F_A = 2^{b(rank(A) - rank(A - I))} = 2^{b(m - rank(A - I))},$$

(5.3)

The equation above shows that the number of fixed points is determined by the rank of the matrix $A - I$, where $A$ is nonsingular. Therefore, it is important to determine the probability distribution of the rank for this type of matrix. Before addressing this, the probability distribution of the rank for a random $m \times m$ matrix over $\mathbb{F}_t$ is considered.

**Rank of Random Matrices**

The probability that an $m \times m$ matrix over $\mathbb{F}_t$ has rank $r$ is $[169, \text{Chap. 3}]$

$$
\hat{p}_{t,m,r} = \frac{1}{t^{mn^2}} \binom{m}{r} \sum_{k=0}^{r} \binom{r}{k} t^{mk+\binom{r-k}{2}}
$$

(5.4)
5.1. Preliminaries

Table 5.2: Theoretical probabilities \( \hat{p}_{2,m,r} \), that \( m \times m \) matrices over \( \mathbb{F}_2 \) have rank \( r \).

<table>
<thead>
<tr>
<th>Rank ( r )</th>
<th>( m )</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>0.375</td>
<td>0.30762</td>
<td>0.28992</td>
<td>0.28879</td>
<td>0.28879</td>
<td>0.28879</td>
<td>0.28879</td>
<td></td>
</tr>
<tr>
<td>( m - 1 )</td>
<td>0.562</td>
<td>0.57678</td>
<td>0.57757</td>
<td>0.57758</td>
<td>0.57758</td>
<td>0.57758</td>
<td>0.57758</td>
<td></td>
</tr>
<tr>
<td>( m - 2 )</td>
<td>0.0625</td>
<td>0.11215</td>
<td>0.12735</td>
<td>0.12835</td>
<td>0.12835</td>
<td>0.12835</td>
<td>0.12835</td>
<td></td>
</tr>
<tr>
<td>( m - 3 )</td>
<td>0</td>
<td>0.00343</td>
<td>0.00512</td>
<td>0.00524</td>
<td>0.00524</td>
<td>0.00524</td>
<td>0.00524</td>
<td></td>
</tr>
<tr>
<td>( m - 4 )</td>
<td>0</td>
<td>1.53E-5</td>
<td>4.41E-5</td>
<td>4.66E-5</td>
<td>4.66E-5</td>
<td>4.66E-5</td>
<td>4.66E-5</td>
<td></td>
</tr>
<tr>
<td>( m - 5 )</td>
<td>0</td>
<td>0</td>
<td>8.6E-8</td>
<td>9.69E-8</td>
<td>9.69E-8</td>
<td>9.69E-8</td>
<td>9.69E-8</td>
<td></td>
</tr>
<tr>
<td>( m - 6 )</td>
<td>0</td>
<td>0</td>
<td>3.79E-11</td>
<td>4.88E-11</td>
<td>4.88E-11</td>
<td>4.88E-11</td>
<td>4.88E-11</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Theoretical probabilities \( \hat{p}_{2^8,m,r} \), that \( m \times m \) matrices over \( \mathbb{F}_{2^8} \) have rank \( r \).

<table>
<thead>
<tr>
<th>Rank ( r )</th>
<th>( m )</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>0.99608</td>
<td>0.99608</td>
<td>0.99608</td>
<td>0.99608</td>
<td>0.99608</td>
<td>0.99608</td>
<td>0.99608</td>
<td></td>
</tr>
<tr>
<td>( m - 1 )</td>
<td>0.00392</td>
<td>0.00392</td>
<td>0.00392</td>
<td>0.00392</td>
<td>0.00392</td>
<td>0.00392</td>
<td>0.00392</td>
<td></td>
</tr>
<tr>
<td>( m - 2 )</td>
<td>2.33E-10</td>
<td>2.33E-10</td>
<td>2.34E-10</td>
<td>2.34E-10</td>
<td>2.34E-10</td>
<td>2.34E-10</td>
<td>2.34E-10</td>
<td></td>
</tr>
<tr>
<td>( m - 3 )</td>
<td>0</td>
<td>2.13E-22</td>
<td>2.13E-22</td>
<td>2.13E-22</td>
<td>2.13E-22</td>
<td>2.13E-22</td>
<td>2.13E-22</td>
<td></td>
</tr>
<tr>
<td>( m - 4 )</td>
<td>0</td>
<td>2.94E-39</td>
<td>2.95E-39</td>
<td>2.95E-39</td>
<td>2.95E-39</td>
<td>2.95E-39</td>
<td>2.95E-39</td>
<td></td>
</tr>
<tr>
<td>( m - 6 )</td>
<td>0</td>
<td>0</td>
<td>2.02E-87</td>
<td>2.02E-87</td>
<td>2.02E-87</td>
<td>2.02E-87</td>
<td>2.02E-87</td>
<td></td>
</tr>
</tbody>
</table>

where the Gaussian coefficients are defined as

\[
\binom{m}{r}_t = \prod_{i=0}^{r-1} \frac{t^{m-i} - 1}{t^{r-i} - 1}.
\]

From Equation 5.4, one can obtain the probability distribution of the rank for a random \( m \times m \) matrix. In this chapter, the focus is only on the case \( t = 2^q \). Calculated probability values for specific values of dimension \( m \) and rank \( r \) for \( q = 1 \) and \( q = 2^8 \) are given in Tables 5.2 and 5.3, respectively. For \( q = 1 \) and \( m \geq 4 \), the probability that the rank \( r \) is less than \( m - 3 \) \((r < m - 3)\) is very low (close to 0). Note that, for \( q = 8 \) and \( m \geq 2 \), the probability that the rank \( r \) is less than \( m - 1 \) \((r < m - 1)\) is also very low. Calculation of the probabilities for these particular values of \( q \) is performed to enable comparison with the probabilities obtained for matrices \( A - I \) examined next.
Table 5.4: Experimental estimates of probabilities of \( r = \text{rank}(A - I) \), where \( A \) is a nonsingular \( m \times m \) matrix over \( \mathbb{F}_{2^q} \)

<table>
<thead>
<tr>
<th>Rank</th>
<th>( q = 1 )</th>
<th>( m = 16 )</th>
<th>( m = 64 )</th>
<th>( m = 128 )</th>
<th>( q = 8 )</th>
<th>( m = 16 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>( m - 1 )</td>
<td>0.288811</td>
<td>0.288814</td>
<td>0.288718</td>
<td>0.277601</td>
<td>0.277754</td>
</tr>
<tr>
<td>( m - 2 )</td>
<td>0.128309</td>
<td>0.128347</td>
<td>0.128345</td>
<td>0.000391</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( m - 3 )</td>
<td>0.005232</td>
<td>0.005237</td>
<td>0.005249</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( m - 4 )</td>
<td>4.69E-5</td>
<td>4.79E-5</td>
<td>4.61E-5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( m - 5 )</td>
<td>9.69E-8</td>
<td>1.34E-7</td>
<td>8.94E-8</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Rank of Matrices of the Type \( A - I \)**

Now consider the probability distribution of the rank of the matrix \( A - I \), where \( A \) is nonsingular. For \( q = 1 \), the nonsingular matrices comprise about 30 percent of all matrices and for \( q = 8 \), they comprise about 99.7 percent. This result is obtained empirically using Magma \(^{[38]}\) as follows.

Experiments were performed for the following four sets of parameter values: \( q = 1 \) and \( m = 16 \), \( q = 1 \) and \( m = 64 \), \( q = 1 \) and \( m = 128 \), \( q = 8 \) and \( m = 16 \). Let \( r = \text{rank}(A - I) \) and let \( T_{m,q}[r] \) denote an array of \( m \) counters. At the start of every experiment, the counters were initialized to zero. The experiment consisted of \( 2^{27} \) trials, where each trial was performed as follows. A random nonsingular matrix \( A \) defined over \( \mathbb{F}_{2^q} \) was generated. Then, the value \( r \) was calculated and the counter \( T_{m,q}[r] \) was incremented by one. At the end of each experiment, the value for each counter \( T_{m,q}[r] \) was divided by \( 2^{27} \) to obtain the probability distribution for selected values of \( r \), given in Table 5.4.

From Table 5.4, for \( q = 1 \), the probabilities that random nonsingular matrices \( A \) have rank \( r \) are approximately the same for different values of \( m \). For \( q = 8 \), the experiment did not generate any matrices with \( r < m - 1 \). The experimental estimate of the probability distribution of the rank for the matrix \( A - I \) obtained here is very similar to the theoretical probability distribution of the rank for a random matrix for the same values of \( q \) and \( m \) (Refer to Tables 5.2 and 5.3 for comparison). Based on this similarity, it is conjectured that both distributions are asymptotically the same.
5.1.3 Linear Diffusion Transformations using Nonsingular Matrices

As defined in Section 2.1.1, the linear diffusion transformation of block ciphers is denoted by \( L \) and the input and output blocks are denoted by \( Z \) and \( X \), respectively. The transformation \( L \) can be represented by the matrix operation given in Equation 5.1 as follows:

\[
X = L(Z) = AZ.
\]

Recall from Equation 5.3 that the number \( F_A \) of fixed points for a transformation \( L \), represented by the matrix \( A \), is determined by the rank of the matrix \( A - I \). Therefore, the results obtained in Section 5.1.2 can be interpreted as follows. For \( q = 1 \) and \( m \geq 4 \), the probability that the value \( F_A \) is greater than \( 2^{3b} \) is very low (close to 0). Similarly, for \( q = 8 \) and \( m \geq 2 \), the probability that \( F_A \) is greater than \( 2^b \) is also very low. These values are used to compare with the results obtained in the analysis of the linear diffusion transformation of several block ciphers presented in Section 5.3.

5.2 Measure of Diffusion Based on Fixed Points

To date, two main criteria have been used for selecting a linear diffusion transformation \( L \), for use in SPN block ciphers [62]. Firstly, the transformation should have relevant diffusion power. Secondly, its performance should be reasonably fast in hardware. In this section, the focus is on the first criterion. The first criterion can be measured by, among others, the avalanche effect, the SAC and the completeness property. However, only the branch number [62, Chap. 9] is focused in this analysis.

The branch number of a linear transformation \( L \), denoted \( B(L) \), is the minimum number of active S-boxes in any two consecutive rounds. The branch number is calculated as follows. Let \( Z = (Z_0, Z_1, \ldots, Z_{m-1}) \) denote a \( mb \)-bit block formed from the concatenation of \( m \) \( b \)-bit words. Let \( \Gamma_Z = (\Gamma_{Z_0}, \Gamma_{Z_1}, \ldots, \Gamma_{Z_{m-1}}) \) denote a binary vector of length \( m \) where \( \Gamma_{Z_i} = 1 \) if \( Z_i \) is nonzero and \( \Gamma_{Z_i} = 0 \) otherwise. Let \( wt(\Gamma_Z) \) denote the Hamming weight (i.e. the number of non-zero

\(^1\)A bijective S-box is active if both its input and output are non-zero in a linear or differential characteristic.
components) of $\Gamma_Z$. The branch number of $L$, denoted $\mathcal{B}(L)$, is defined as

$$\mathcal{B}(L) = \min \{ \text{wt}(\Gamma_Z) + \text{wt}(\Gamma_X) : Z \neq 0 \text{ and } X = L(Z) \}$$  \hspace{1cm} (5.5)$$

where $\mathcal{B}(L) \leq m + 1$.

The branch number of $L$ is optimal if $\mathcal{B}(L) = m + 1$ and the linear code associated with $L$ has the highest minimum distance. The reader is referred to MacWilliams and Sloane [130] for a thorough discussion about linear codes. In order to obtain the highest minimum distance of particular linear codes, the online databases provided by Grassl [81]; and Schmid and Schürer [157] were consulted.

The proposed new measure of diffusion is based on the existence of fixed points and of simple linear relationships between the input and output blocks. Let $I(l) = [\alpha_{(l)ij}]$ denote the $m \times m$ matrix based on the identity matrix $I = [\alpha_{ij}]$ where $I(0) = I$. The elements of the matrix $I(l)$ are determined by the rotation parameter $l \in \{0, 1, \ldots, m - 1\}$ where $\alpha_{(l)ij} = \alpha_{i,(j-l)\mod m}$. The following are examples of the matrices $I(1)$ and $I(2)$ for $m = 4$.

\[
I(1) = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0
\end{pmatrix}, \quad I(2) = \begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{pmatrix}.
\]

Let $A$ denote the matrix form of the linear transformation $L$. The proposed measure considers input blocks that have the following simple relationship which is an extension of Equation 5.2

\[
(A - I(l))Z = 0.
\]  \hspace{1cm} (5.6)

where $l \in \{0, 1, \ldots, m - 1\}$. For $l = 0$, the solution to the above linear equations gives the set of all fixed points for $L$, which is the same as Equation 5.2. For $l > 0$, the solution to the above linear relationship gives the set of all input blocks that are only rotated $lb$ bits to the left by the linear transformation $L$ to produce the output blocks. This relationship is given as follows where $\hat{Z}$ represents particular input blocks to $L$:

\[
L(\hat{Z}) = \hat{Z} \ll lb.
\]  \hspace{1cm} (5.7)
The number of input blocks that satisfy the above relationship is calculated as:

\[ \mathcal{F}_{A(l)} = 2^{b(m - \text{rank}(A - I(l)))} \quad (5.8) \]

where \( l \in \{0, 1, \ldots, m - 1\} \).

The proposed new measure of diffusion is given by the number \( \mathcal{D}(A) \) defined as follows:

\[
\mathcal{D}(A) = \frac{1}{m^{2mb}} \sum_{l=0}^{m-1} \mathcal{F}_{A(l)} = \frac{1}{m^{2mb}} \sum_{l=0}^{m-1} 2^{b(m - \text{rank}(A - I(l)))}
\]

where \( 2^{-mb} \leq \mathcal{D}(A) \leq 1 \) and \( A \) represents the matrix form of the linear transformation \( L \). The number \( \mathcal{D}(A) \) denotes the average fraction of input blocks to \( L \) that have the linear relationship of the form given by Equation (5.6). A large \( \mathcal{D}(A) \) value means that there are many input blocks that are effectively unchanged by the linear transformation when producing the output blocks. Likewise, a low \( \mathcal{D}(A) \) number means that there are many input blocks that are effectively changed by the linear transformation when producing the output blocks.

In the context of the \( \mathcal{D}(A) \) number, the relevant diffusion power is related to how well the transformation effectively changes the value of the input block when producing the output block. Using this measure of diffusion, it is clear that no diffusion exists at the fixed points. Therefore, the transformation has poor diffusion for these particular input blocks. The existence of the linear relationship given in Equation (5.6) also implies that for particular input words, the linear transformation has a much simpler representation. In other words, the linear transformation of the SPN block cipher can be replaced by Equation (5.6) for an average of \( \mathcal{D}(A) \cdot 2^{mb} \) input blocks.

### 5.3 Applications

This section presents the application of the proposed new measure of diffusion to the linear transformation of the following SPN block ciphers: the AES [62], ARIA [116, 146], Present [37] and Serpent [4]. The descriptions of the transformations \( L \) for these ciphers are given in Sections 2.4.1, 2.4.2, 2.4.5 and 2.4.4, respectively. The \( \mathcal{D}(A) \) numbers and the number of fixed points for these linear transformations are calculated. A summary of the findings is presented in
Table 5.5: Parameters of $L$

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Matrix $A$</th>
<th>Input-Output</th>
<th>Block size (mb bits)</th>
<th>Branch number</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>16 $F_{28}$</td>
<td>$F_{28}$</td>
<td>128</td>
<td>5</td>
</tr>
<tr>
<td>ARIA</td>
<td>16 $F_2$</td>
<td>$F_{28}$</td>
<td>128</td>
<td>8</td>
</tr>
<tr>
<td>Present</td>
<td>64 $F_2$</td>
<td>$F_2$</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>Serpent</td>
<td>128 $F_2$</td>
<td>$F_2$</td>
<td>128</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 5.5 gives the parameters of $L$ for the SPN ciphers examined in this chapter. Note that only the AES and ARIA explicitly use branch number as a measure of diffusion. Both Present and Serpent use a different measure of diffusion, however, the branch numbers for these ciphers can be easily computed. The computation is trivial for Present. For Serpent, it is necessary to find the bits that activate the fewest number of active S-boxes over two rounds. One example is as follows. If bit 56 is non-zero and the rest are zero, then two S-boxes in the next round will have one non-zero input bit each, that is S-boxes 25 (in the second bit) and 30 (in the first bit). Refer to Table 2.3 in Section 2.4.4 for the position of these bits.

### 5.3.1 AES

The transformation $L$ of the AES has $F_{A(0)} = 2^{16}$ fixed points which have the following form where $Z_i \in F_{28}$:

\[
Z_i \oplus 8DZ_{14} \oplus 8CZ_{15} = 0 \quad \text{for } i = 0, 5, 8, 13, \quad Z_i \oplus Z_{15} = 0 \quad \text{for } i = 2, 7, 10,
\]
\[
Z_i \oplus 8CZ_{14} \oplus 8DZ_{15} = 0 \quad \text{for } i = 1, 4, 9, 12, \quad Z_i \oplus Z_{14} = 0 \quad \text{for } i = 3, 6, 11.
\]

On average, there are $2^{15.0056}$ input blocks that satisfy the linear relationship of the form given in Equation 5.6. Exact values of $F_{A(l)}$ for $l \in \{0, 1, \ldots, 15\}$ are given in Table 5.6.
Table 5.6: Number of input blocks to the transformation \( L \) of the AES such that \( AZ = I(l)Z \)

<table>
<thead>
<tr>
<th>( l )</th>
<th>( \mathcal{F}_{A(i)} )</th>
<th>( \mathcal{F}_{A(i)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2(^{16})</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2(^{8})</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2(^{16})</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2(^{8})</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>2(^{16})</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>2(^{8})</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>2(^{16})</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>2(^{8})</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 5.7: Number of input blocks to the transformation \( L \) of ARIA such that \( AZ = I(l)Z \)

<table>
<thead>
<tr>
<th>( l )</th>
<th>( \mathcal{F}_{A(i)} )</th>
<th>( \mathcal{F}_{A(i)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2(^{72})</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2(^{24})</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2(^{24})</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2(^{16})</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>2(^{32})</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>2(^{8})</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>2(^{24})</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>2(^{16})</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 5.8: Number of input blocks to the transformation \( L \) of PRESENT such that \( AZ = I(l)Z \)

<table>
<thead>
<tr>
<th>( l )</th>
<th>( \mathcal{F}_{A(i)} )</th>
<th>( \mathcal{F}_{A(i)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2(^{24})</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2(^{7})</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>2(^{6})</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>2(^{21})</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>2(^{4})</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>2(^{5})</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>2(^{18})</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>2(^{5})</td>
<td>15</td>
</tr>
</tbody>
</table>

5.3.2 ARIA

The transformation \( L \) of ARIA has \( \mathcal{F}_{A(0)} = 2^{72} \) fixed points which have the following form where \( Z_i \in \mathbb{F}_{2^8} \):

\[
\begin{align*}
Z_0 \oplus Z_3 \oplus Z_6 \oplus Z_7 \oplus Z_9 \oplus Z_{11} &= 0, \\
Z_2 \oplus Z_3 \oplus Z_{10} \oplus Z_{11} \oplus Z_{14} \oplus Z_{15} &= 0, \\
Z_4 \oplus Z_7 \oplus Z_9 \oplus Z_{10} \oplus Z_{13} \oplus Z_{14} &= 0, \\
Z_5 \oplus Z_6 \oplus Z_9 \oplus Z_{10} \oplus Z_{11} \oplus Z_{14} &= 0, \\
Z_8 \oplus Z_7 \oplus Z_9 \oplus Z_{10} \oplus Z_{11} &= 0, \\
Z_{12} \oplus Z_{13} \oplus Z_{14} \oplus Z_{15} &= 0, \\
Z_1 \oplus Z_3 \oplus Z_6 \oplus Z_7 \oplus Z_9 \oplus Z_{10} \oplus Z_{14} \oplus Z_{15} &= 0.
\end{align*}
\]

On average, there are \( 2^{68} \) input blocks that satisfy the linear relationship of the form given in Equation 5.6. Exact values of \( \mathcal{F}_{A(l)} \) for \( l \in \{0, 1, \ldots, 15\} \) are given in Table 5.7.
5.3.3 PRESENT

The transformation $L$ of PRESENT has $\mathcal{F}_{A(0)} = 2^{24}$ fixed points which have the following form where $Z_i \in \mathbb{F}_2$:

\[
\begin{align*}
Z_1 &= Z_4 = Z_{16}, & Z_2 &= Z_8 = Z_{32}, & Z_3 &= Z_{12} = Z_{48}, & Z_5 &= Z_{17} = Z_{20}, \\
Z_6 &= Z_{24} = Z_{33}, & Z_7 &= Z_{28} = Z_{49}, & Z_9 &= Z_{18} = Z_{36}, & Z_{10} &= Z_{34} = Z_{40}, \\
Z_{11} &= Z_{44} = Z_{50}, & Z_{13} &= Z_{19} = Z_{52}, & Z_{14} &= Z_{35} = Z_{56}, & Z_{15} &= Z_{51} = Z_{60}, \\
Z_{22} &= Z_{25} = Z_{37}, & Z_{23} &= Z_{29} = Z_{53}, & Z_{26} &= Z_{38} = Z_{41}, & Z_{27} &= Z_{45} = Z_{54}, \\
Z_{30} &= Z_{39} = Z_{57}, & Z_{31} &= Z_{55} = Z_{61}, & Z_{43} &= Z_{46} = Z_{58}, & Z_{47} &= Z_{59} = Z_{62}.
\end{align*}
\]

On average, there are $2^{18.3645}$ input blocks that satisfy the linear relationship of the form given in Equation 5.6. Exact values of $\mathcal{F}_{A(l)}$ for $l \in \{0, 1, \ldots, 63\}$ are given in Table 5.8. Note that the bits $Z_0$, $Z_{21}$, $Z_{42}$ and $Z_{63}$ are absent from the above equations. These 4 bits correspond to the bits that are invariant under transformation $L$.

5.3.4 Serpent

The transformation $L$ of Serpent only has $\mathcal{F}_{A(0)} = 2$ fixed points given as follows:

\[
\begin{align*}
00000000 & \quad 00000000 & \quad 00000000 & \quad 00000000, \\
909C6ACE & \quad D3096AD6 & \quad 6D71BEA0 & \quad 9267AAC9.
\end{align*}
\]

On average, there are $2^{1.0444}$ input blocks that satisfy the linear relationship of the form given in Equation 5.6. Exact values of $\mathcal{F}_{A(l)}$ for $l \in \{0, 1, \ldots, 127\}$ are given in Table 5.9.

5.3.5 Analysis

Table 5.10 summarizes the results of the application of the proposed new method of measuring diffusion to the block ciphers the AES, ARIA, PRESENT and Serpent. For each cipher, the dimension $m$ of the matrix $A$ and the rank $r = \text{rank}(A - I)$ of the matrix $A - I$ are given. The probability $\hat{p}_{2^{4},m,r}$, that a randomly chosen matrix has the same rank is also given. Table 5.10 also shows, for each linear transformation, the number $\mathcal{D}(A)$ and the number $\mathcal{F}_{A(0)}$. 
Table 5.9: Number of input blocks to the transformation $L$ of Serpent such that $AZ = I_{(l)}Z$

<table>
<thead>
<tr>
<th>$l$</th>
<th>$\mathcal{F}_{A(0)}$</th>
<th>$l$</th>
<th>$\mathcal{F}_{A(0)}$</th>
<th>$l$</th>
<th>$\mathcal{F}_{A(0)}$</th>
<th>$l$</th>
<th>$\mathcal{F}_{A(0)}$</th>
<th>$l$</th>
<th>$\mathcal{F}_{A(0)}$</th>
<th>$l$</th>
<th>$\mathcal{F}_{A(0)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>16</td>
<td>2</td>
<td>32</td>
<td>2</td>
<td>48</td>
<td>2</td>
<td>64</td>
<td>2</td>
<td>80</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>17</td>
<td>2</td>
<td>33</td>
<td>2</td>
<td>49</td>
<td>2</td>
<td>65</td>
<td>2</td>
<td>81</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>18</td>
<td>2</td>
<td>34</td>
<td>2</td>
<td>50</td>
<td>2</td>
<td>66</td>
<td>2</td>
<td>82</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>19</td>
<td>2</td>
<td>35</td>
<td>2</td>
<td>51</td>
<td>2</td>
<td>67</td>
<td>2</td>
<td>83</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>20</td>
<td>2</td>
<td>36</td>
<td>2</td>
<td>52</td>
<td>2</td>
<td>68</td>
<td>2</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>21</td>
<td>2</td>
<td>37</td>
<td>2</td>
<td>53</td>
<td>2</td>
<td>69</td>
<td>2</td>
<td>85</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>22</td>
<td>2</td>
<td>38</td>
<td>2</td>
<td>54</td>
<td>2</td>
<td>70</td>
<td>2</td>
<td>86</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>23</td>
<td>2</td>
<td>39</td>
<td>2</td>
<td>55</td>
<td>2</td>
<td>71</td>
<td>2</td>
<td>87</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>24</td>
<td>2</td>
<td>40</td>
<td>2</td>
<td>56</td>
<td>2</td>
<td>72</td>
<td>2</td>
<td>88</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>25</td>
<td>2</td>
<td>41</td>
<td>2</td>
<td>57</td>
<td>2</td>
<td>73</td>
<td>2</td>
<td>89</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>26</td>
<td>2</td>
<td>42</td>
<td>2</td>
<td>58</td>
<td>2</td>
<td>74</td>
<td>2</td>
<td>90</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>27</td>
<td>2</td>
<td>43</td>
<td>2</td>
<td>59</td>
<td>2</td>
<td>75</td>
<td>2</td>
<td>91</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>28</td>
<td>2</td>
<td>44</td>
<td>2</td>
<td>60</td>
<td>2</td>
<td>76</td>
<td>2</td>
<td>92</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>29</td>
<td>2</td>
<td>45</td>
<td>2</td>
<td>61</td>
<td>2</td>
<td>77</td>
<td>2</td>
<td>93</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>30</td>
<td>2</td>
<td>46</td>
<td>2</td>
<td>62</td>
<td>2</td>
<td>78</td>
<td>2</td>
<td>94</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>31</td>
<td>2</td>
<td>47</td>
<td>2</td>
<td>63</td>
<td>2</td>
<td>79</td>
<td>2</td>
<td>95</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.10: Summary of the observations on the transformation $L$

<table>
<thead>
<tr>
<th>Cipher</th>
<th>$m$</th>
<th>$r$</th>
<th>$\hat{p}_{2^q,m,r}$</th>
<th>$D(A)$</th>
<th>$\mathcal{F}_{A(0)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>16</td>
<td>14</td>
<td>2.34E-10</td>
<td>$2^{-112.9944}$</td>
<td>$2^{16}$</td>
</tr>
<tr>
<td>ARIA</td>
<td>16</td>
<td>7</td>
<td>1.42E-24</td>
<td>$2^{-60}$</td>
<td>$2^{72}$</td>
</tr>
<tr>
<td>PRESENT</td>
<td>64</td>
<td>40</td>
<td>1.40E-173</td>
<td>$2^{-45.6355}$</td>
<td>$2^{24}$</td>
</tr>
<tr>
<td>Serpent</td>
<td>128</td>
<td>127</td>
<td>0.57758</td>
<td>$2^{-126.9556}$</td>
<td>2</td>
</tr>
</tbody>
</table>

The results in Table 5.10 clearly show that for all of the ciphers examined, the matrices $A - I$ do not have full rank. For $q = 1$, the matrix $A - I$ of Serpent has rank $r = m - 1$, ARIA has $r = m - 9$ and PRESENT has $r = m - 24$. For $q = 8$, the matrix $A - I$ of the AES has rank $r = m - 2$. It is known from Section 5.1.2 that for a randomly chosen matrix with $q = 1$, it is unlikely that the rank $r$ is less than $m - 3$ ($r < m - 3$) and for $q = 8$, it is unlikely that the rank $r$ is less than $m - 1$ ($r < m - 1$). Therefore, only the matrix $A - I$ of Serpent has rank within the expectation for a random matrix. This is reflected in the high probability value $\hat{p}_{2^q,m,r}$ for Serpent, compared to the other three ciphers.

Similarly, for $q = 1$, the transformation $L$ of Serpent has $\mathcal{F}_{A(0)} = 2^b$ fixed points, ARIA has $\mathcal{F}_{A(0)} = 2^{2b}$ and PRESENT has $\mathcal{F}_{A(0)} = 2^{24b}$. For $q = 8$, the
transformation $L$ of the AES has $F_{A(0)} = 2^{2b}$ fixed points. Refer to Table 5.5 for the specific values of $b$ for each of these ciphers. Recall from Section 5.1.3 that for $q = 1$, it is unlikely that $F_{A(0)} > 2^{3b}$ and for $q = 8$, it is unlikely that $F_{A(0)} > 2^b$. Consequently, only the transformation $L$ of Serpent has a number of fixed points within the expectation for a random linear transformation.

It can also be noted from Table 5.10 that the block cipher Present has the largest $D(A)$ number, followed by ARIA. This means that the linear transformation of Present has the largest fraction of output blocks that are just simple linear rotations of the input block. So, there are many input blocks that are not effectively changed by the linear transformation of Present when producing the output block. Both the AES and Serpent have very small $D(A)$ numbers, which indicate that only a small fraction of the output blocks is produced by simple rotations of the input block.

The above observations motivate the following claims. Firstly, the linear transformations $L$ for all ciphers except for Serpent do not behave as expected for random linear transformations. The excessively small probabilities $\hat{p}_{2q,m,r}$ given in Table 5.10 support this argument. Secondly, for these transformations, there exist a large number of inputs for which the transformation provides no diffusion. The relatively high number $F_{A(0)}$ of fixed points substantiate this claim.

### 5.4 Fixed Points and Existing Design Criteria

Recall from Section 5.2 that the existing design criteria for the linear transformation $L$ used in SPN block ciphers are optimal branch number and reasonably high performance. The branch numbers of $L$ for the AES, ARIA, Present and Serpent are 5, 8, 2 and 3, respectively; whereas the optimal branch numbers of $L$ for these ciphers are 17, 8, 8 (if $m = 16$ and $q = 1$) and 12 (if $m = 32$ and $q = 1$) [81, 157], respectively. Note that the optimal branch numbers calculated here are with respect to the number of S-boxes in the round function of the ciphers such that the transformation $L$ operates word-wise. For example, the AES’ MixColumns transformation has branch number 5, which is optimal for a $4 \times 4$ matrix which operates word-wise on four S-boxes at a time. However, it is not optimal if the transformation operates on all 16 S-boxes at once. To simplify the calculation of the optimal branch numbers for the transformations $L$
of both Present and Serpent, it is assumed that the input and output vectors are defined over $\mathbb{F}_{2^4}$ (i.e. the transformation $L$ operates word-wise, instead of bit-wise).

The results presented in Section 5.3 do not indicate a clear relationship between existing design criteria and the number of fixed points. For instance, the transformation $L$ of ARIA has optimal branch number but considerably many fixed points. The transformation $L$ of the AES has low branch number and high number of fixed points. For Serpent, despite having a relatively low branch number, the transformation $L$ has very few fixed points. Finally, the transformation $L$ for Present has very low branch number and an extremely large number of fixed points.

In the remainder of this subsection, the relationship between the branch number, number of fixed points and performance is investigated by considering the effect of replacing matrix $A$ by an alternate matrix $B$. The aim is to find a matrix $B$ which has fewer fixed points and/or higher branch number, but without sacrificing on performance.

### 5.4.1 AES

The transformation $L$ of the AES has branch number 5 and a reasonably fast performance \cite{52}. The performance aspect is achieved by restricting each entry of the matrix $A$ to be a small number in $\mathbb{F}_{2^8}$. Furthermore, the designers did not choose a transformation with optimal branch number 17 due to performance considerations. Grošek and Zając \cite{83} found a different matrix $B$ with small-number entries but fairly large-number entries for its inverse, $B^{-1}$. This matrix
Chapter 5. Diffusion in the Linear Transformations of SPN Block Ciphers

$B$ is given below.

$$
B = \begin{pmatrix}
3 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\
0 & 1 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\
0 & 3 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\
0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 \\
0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\
0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{pmatrix}
$$

Therefore, the transformation that uses this matrix $B$ is faster in encryption than decryption. The matrix $B - I$ has rank 16 which corresponds to one fixed point. If the matrix $B$ is used in the transformation $L$ of the AES, then the number of fixed points is reduced by a factor of $2^{16}$, from $F_{A(0)} = 2^{16}$ to $F_{B(0)} = 1$. In order to have optimal branch number 17, Nakahara and Abrahão [137] proposed a symmetric matrix (an involution), denoted $B'$ where the rank of $B' - I$ is 8. This matrix $B'$ is given as follows.

$$
B' = \begin{pmatrix}
1 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & A & B & C & D & E & 10 & 2 & 1E \\
3 & 1 & 5 & 4 & 7 & 6 & 9 & 8 & B & A & D & C & 10 & E & 1E & 2 \\
4 & 5 & 1 & 3 & 9 & 6 & 7 & C & D & A & B & 2 & 1E & E & 10 & 1E \\
5 & 4 & 3 & 1 & 9 & 8 & 7 & 6 & D & C & B & A & 1E & 2 & 10 & E \\
6 & 7 & 8 & 9 & 1 & 3 & 4 & 5 & E & 10 & 2 & 1E & A & B & C & D \\
7 & 6 & 9 & 8 & 3 & 1 & 5 & 4 & 10 & E & 1E & 2 & B & A & D & C \\
8 & 9 & 6 & 7 & 4 & 5 & 1 & 3 & 2 & 1E & E & 10 & C & D & A & B \\
9 & 8 & 7 & 6 & 5 & 4 & 3 & 1 & 1E & 2 & 10 & E & D & C & B & A \\
A & B & C & D & E & 10 & 2 & 1E & 1 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
B & A & D & C & 10 & E & 1E & 2 & 1E & 2 & 3 & 1 & 5 & 4 & 7 & 6 & 9 & 8 \\
C & D & A & B & 2 & 1E & E & 10 & 4 & 5 & 1 & 3 & 8 & 9 & 6 & 7 \\
D & C & B & A & 1E & 2 & 10 & E & 5 & 4 & 3 & 1 & 9 & 8 & 7 & 6 \\
E & 10 & 2 & 1E & A & B & C & D & 6 & 7 & 8 & 9 & 1 & 3 & 4 & 5 \\
10 & E & 1E & 2 & B & A & D & C & 7 & 6 & 9 & 8 & 3 & 1 & 5 & 4 \\
2 & 1E & E & 10 & C & D & A & B & 8 & 9 & 6 & 7 & 4 & 5 & 1 & 3 \\
1E & 2 & 10 & E & D & C & B & A & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 1 \\
\end{pmatrix}
$$

If this matrix $B'$ replaces $A$, then the number of fixed points is significantly increased by a factor of $2^{48}$, from $F_{A(0)} = 2^{16}$ to $F_{B'} = 2^{64}$. 
5.4.2 ARIA

The transformation $L$ of ARIA is designed to satisfy two main properties: branch number 8 and involution \[114\]. Using the method explained by the designers \[114\], a search for other matrices $B$ that have rank of $B - I$ higher than 7 was conducted. As a result, 2376 different matrices were found that maintain the two main properties, and for which the rank of the matrix $B - I$ is 8. An example of such a matrix $B$ is given below.

$$
B = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0
\end{pmatrix}
$$

If this matrix $B$ replaces $A$ in the transformation $L$ of ARIA, the number of fixed points is reduced by a factor of $2^8$, from $\mathcal{F}_{A(0)} = 2^{72}$ to $\mathcal{F}_{B(0)} = 2^{64}$. It is believed that the performance of this new transformation is comparable to the original transformation because both only use XOR operations, which are extremely fast in hardware.

5.4.3 PRESENT

The transformation $L$ of PRESENT is designed primarily for efficiency and simplicity in security analysis \[37\]; hence the choice of a bit permutation. As before, a search was conducted for other matrices $B$ such that the rank of $B - I$ is larger than 40. The constraint that the output bits go to the same group of S-boxes outlined in the specification \[1\] was applied. One way of preserving this property is by rotating the bits only within each group. As a result, a matrix $B$ was found such that the rank of $B - I$ is 61. The matrix $B$ is given in Table 5.11. If this

\[\text{Therefore, the search is limited to transformations having branch number 2.}\]
Table 5.11: Permutation of bits in the modified $L$ transformation of Present. $P(i)$ means bit $i$ is moved to position $P(i)$ after the transformation.

<table>
<thead>
<tr>
<th>$i$</th>
<th>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(i)$</td>
<td>48 0 16 32 33 49 1 17 18 34 50 2 3 19 35 51</td>
</tr>
<tr>
<td>$i$</td>
<td>16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</td>
</tr>
<tr>
<td>$P(i)$</td>
<td>52 4 20 36 37 53 5 21 22 38 54 6 7 23 39 55</td>
</tr>
<tr>
<td>$i$</td>
<td>32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47</td>
</tr>
<tr>
<td>$P(i)$</td>
<td>56 8 24 40 41 57 9 25 26 42 58 10 11 27 43 59</td>
</tr>
<tr>
<td>$i$</td>
<td>48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63</td>
</tr>
<tr>
<td>$P(i)$</td>
<td>60 12 28 44 45 61 13 29 30 46 62 14 63 15 31 47</td>
</tr>
</tbody>
</table>

Matrix $B$ replaces $A$, then the number of fixed points is significantly reduced by a factor of $2^{21}$, from $F_{A(0)} = 2^{24}$ to $F_{B(0)} = 2^3$. Note that the value $F_{B(0)}$ is within the expectation for a random linear transformation.

### 5.4.4 Serpent

The transformation $L$ of Serpent is designed to meet three criteria: high avalanche, simplicity and reasonable security bounds with respect to differential and linear cryptanalysis [4]. The method used to construct $L$ is not explicitly outlined in the specification. Thus, a search was conducted for other matrices that have a larger branch number than the matrix $A$. Recall that the matrix $A$ used in Serpent has branch number 3. In order to easily calculate this number for the proposed matrix $B$, the input block was regarded as thirty-two 4-bit words. These words correspond to the outputs of the preceding S-boxes. The $32 \times 32$ nonsingular matrix $B$ over $\mathbb{F}_2$ designed by Koo, Jang and Song [115] has branch number 10,
and is as follows.

\[
B = \begin{pmatrix}
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 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 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}

The rank of the matrix \( B - I \) is 26 and using this matrix \( B \), the number of fixed points is increased by a factor of \( 2^{23} \), from \( F_{A(o)} = 2 \) to \( F_{B(o)} = 2^{24} \). In this case, the increase in branch number by replacing \( A \) with \( B \) results in a significant increase in the number of fixed points.

### 5.4.5 Branch Number, Fixed Points and Performance

The findings in this section suggest that links exist between the branch number, number of fixed points and performance of a transformation \( L \). However, the existence of a specific relationship is unclear. For the AES, the decrease in the number of fixed points comes at the expense of slow decryption. Alternatively, the increase in the branch number resulted in the increase in the number of fixed points. Due to the design properties of ARIA, an alternate matrix that
slightly lowers the number of fixed points was identified, but it is still higher than
the number for a random linear transformation. The simple design criteria for
PRESENT prohibits an increase in the branch number; nonetheless, an alternate
matrix was found that significantly decreases the number of fixed points. For
Serpent, an alternate matrix that has a higher branch number than the original
matrix consequently has more fixed points.

In the investigation, a transformation $L$ that simultaneously has a high branch
number, low number of fixed points and reasonably high performance was unable
to be found. Since for most ciphers, an exhaustive search was not conducted, such
a matrix might exist. As a cautionary note, the use of the alternative matrices
proposed in this section is not recommended until further security analysis is
performed.

5.5 Cryptographic Significance

For SPN ciphers, the existence of fixed points in the transformation $L$ hints
at the presence of 1-round self-iterating differential characteristics. Note that
not all fixed points are useful in constructing a self-iterating characteristic. The
usefulness of a fixed point, in this case, depends on its interaction with the
subsequent nonlinear transformation. If the input difference $\Delta Z$ is a fixed point,
then the transformation $L$ will replicate this difference into the same S-boxes in
the next round. For instance, there exists a 1-round self-iterating characteristic
for ARIA where only the four left-most S-boxes are active. Biryukov et al. [29]
show this characteristic on a previous variant of ARIA. If the four left-most bytes
are identical, then it is a fixed point for the transformation $L$.

Based on the Difference Distribution Table for the S-boxes of ARIA, we ad-
ditionally show that there are 3976 possible 1-round self-iterating characteristics
with average probability of $2^{-28}$. This 1-round self-iterating characteristic can
be concatenated up to 4 rounds with probability $(2^{-28})^4 = 2^{-112}$. The character-
istic can be exploited in a similar attack demonstrated by Biryukov et al. [29].
Similarly, by exhaustive search, three 1-round self-iterating characteristics for
PRESENT were found. The characteristic involves 4 active S-boxes per round
with a 1-round probability of $(2^{-3})^4 = 2^{-12}$. The characteristic exploits some
poor diffusion of the transformation $L$ of PRESENT noted by Collard and Stan-
daert [47]. However, this 1-round self-iterating characteristic can only be con-
cated up to 5 out of 31 rounds of Present. In contrast, the best 14-round
differential characteristic uses only two active S-boxes per round [174].

Other than 1-round self-iterating characteristics, the existence of fixed points
in the transformation $L$ of SPN ciphers might also be useful to an attacker if
the S-boxes in the nonlinear transformation $S$ of the round function also have
the same fixed points. The fixed points in these two transformations may lead
to fixed points in the round function as follows. If $X$ is a fixed point for both $S$
and $L$, then $X = L(S(X))$. If there is more than one fixed point in $L(S(\cdot))$, then
the round function does not appear random, which might indicate a weakness.
Serpent has the least number of fixed points in the transformation $L$, compared
to other ciphers. Interestingly, 5 out of 8 S-boxes of Serpent have at least one
fixed point. Despite this, due to the lack of fixed points in the transformation
$L$, there is no trivial way to exploit the fixed points in the transformation $S$. In
contrast, the S-boxes of the other block ciphers examined in this chapter do not
have fixed points.

Furthermore, the transformations $L$ in the round function of most SPN block
Ciphers are preceded by a key addition and a nonlinear transformation. Note that
the integral attack [34, 112, 129] is applicable to any word- and bit-based [180]
SPN ciphers, provided that the S-boxes used in the nonlinear transformation are
bijective. In this case, apart from the number of rounds, the transformation $L$
plays a crucial role in defending this type of attack. Therefore, the existence of
fixed points might be possibly exploited in advanced variants of this attack.

5.6 Related Work

In a related work, Song and Seberry [165] study the cycle length of each trans-
formation in the round function of the AES. They showed that if the input bytes
of the transformation $L$ are identical, then it has a cycle of length one. In other
words, the $2^8$ inputs of the form $Z_i = Z_{i+1}$ for $i = 0, 1, \ldots, 14$ and $Z_i \in F_{2^8}$
are fixed points. Song and Seberry do not explicitly mention other input forms
that have cycle length one. Results in this chapter show that there are $2^{16}$ fixed
points. So clearly, there are $2^{16} - 2^8$ additional inputs which also have cycle
length one.

Recently, Collard and Standaert [17] observed a weakness in the transformation
$L$ of Present. They noted that certain input bits of the transformation
are moved to the same set of S-boxes in the next round. This observation is related to the existence of fixed points. As an example, they fixed the values of the bits \( Z_{22} = Z_{25} = Z_{37}, Z_{36} = Z_{38} = Z_{41}, Z_{21} \) and \( Z_{42} \) where \( Z_i \in \mathbb{F}_2; \) and observed a non-uniform distribution across some rounds. If the linear diffusion transformation of \textsc{Present} is replaced with the matrix proposed in Section 5.4, then this weakness is eliminated.

5.7 Conclusion

In this chapter, a new method of measuring the diffusion provided by the linear transformation of SPN block ciphers is proposed. The measure denotes the average number of input blocks that are effectively unchanged by the linear transformation when producing the output blocks. It was also shown that the linear diffusion transformations of all of the SPN block ciphers examined in this chapter except Serpent, do not behave as expected for random linear transformations. This is because the number of fixed points in these transformations greatly exceed the expected number for a random linear transformation. The transformations provide poor diffusion for many input blocks since the bits in these blocks are unchanged when producing the output blocks.

This work also suggests that a connection exists between the number of fixed points and existing design criteria for \( L \), although the exact relationship remains unclear. This was explored by investigating alternate matrices for \( L \). For instance, increasing the branch number may also increase the number of fixed points. This was the case for the alternate matrices for the AES and Serpent. However, matrices were found which had the same branch number, but reduced the number of fixed points for AES, ARIA and \textsc{Present}. For the AES, this additionally comes at the expense of slow decryption. For any of the block ciphers examined, the search for a transformation \( L \) that has all the following three properties were unsuccessful: an optimal branch number, reasonably high performance and a number of fixed points within the expectation for a random linear transformation. It is therefore an open problem whether such transformations exist.

Additionally, possible techniques were explored to exploit the existence of fixed points in cryptanalysis. Although no major attack that exploits the fixed points was found, it is believed that they are still an important property that
should be considered. It is stressed that it is not the intent to replace the branch number, rather to complement it. Hence, apart from branch numbers, one should also take into account the number of fixed points as one of the design criteria when selecting linear diffusion transformations for SPN block ciphers.
Chapter 6

Linearity within the SMS4 Block Cipher

The SMS4 block cipher [64, 149] is a standard block cipher used in the Chinese Wireless LAN Wired Authentication and Privacy Infrastructure (WAPI). Reduced-round versions of the cipher have been cryptanalyzed using integral [125], rectangle [104, 126, 168, 183], impossible differential [126, 168], boomerang [104], differential [104, 183] and linear [74, 104] attacks. The best attack so far is a differential attack on 22 rounds by Zhang et al. [185]. In the same paper, they observe that the number of rotations and XOR operations used in the linear transformation of the SMS4 block cipher are the minimum required to achieve an optimal branch number. They also show that the linear transformation is bijective and present the distribution of input and output patterns of this transformation to assist in differential attacks.

In this chapter, the results of new analysis on linearity within the components of the SMS4 block cipher is presented. In particular, every linear and nonlinear component is examined to determine whether there exist input words that are either fixed points or just rotations of the output words. This linear relationship is the same relationship that the new proposed measure of diffusion for linear transformations of SPN block ciphers is based on (presented in Chapter 5). However, in this chapter, for SMS4, in addition to linear transformations, the nonlinear transformations, which do not have equivalent matrix representations, are also examined. Furthermore, Chapter 5 focused exclusively on SPN block
ciphers, whereas SMS4 is a Feistel-type block cipher.

The results presented in this chapter are new and show some peculiarities inherent in the components of the SMS4 block cipher. The crucial result is the existence of fixed points and also of simple linear relationships between the bits of the input and output words for each component in the round functions. In particular, it is shown that the nonlinear function $T$ has 11 fixed points, which is more than the expected number for a random permutation. Therefore, the behaviour of the function $T$ does not appear random. This work also identifies a set of input words for which the round functions of both the encryption and the key scheduling algorithms produce the same output words. Furthermore, it is shown that the branch number of the linear transformation in the key scheduling algorithm is four, which is less than optimal.

One of the implications of these observations is that, for all keys, the first four round functions of SMS4 are not always nonlinear. Under this condition, the number of effective rounds is reduced by four: from 32 to 28. A brief exploration of the susceptibility of SMS4 against algebraic and advanced variants of the slide attacks is conducted. Finally, it is demonstrated that if the linear transformation in the key scheduling algorithm is used in the encryption algorithm, then this variant of SMS4, reduced to 27 rounds, is vulnerable to a differential attack. In contrast, the best differential attack on the original SMS4 is on 22 rounds \[185\], which is also the best existing attack so far. These observations might potentially be useful in attacking SMS4 itself.

This chapter is organized as follows. The observations on the components in the round functions of both the encryption and the key scheduling algorithms of the SMS4 block cipher are presented in Section 6.1. Section 6.2 discusses the cryptographic significance of these observations. Section 6.3 presents a differential attack on a slightly modified variant of SMS4. A summary of the observations and conclusions are given in Section 6.4.

### 6.1 Observations on Components in the Round Functions

This section presents several new observations on each component in the round functions of both the encryption and the key scheduling algorithms of SMS4.

\[1\] Refer to Section 5.1 for some preliminaries about fixed points in random permutations.
The description of SMS4 is given in Section 2.4.6.

6.1.1 Simple Linear Relationships between Input and Output Words

The existence of a simple linear relationship between the bits of some input and output words of each component in the encryption and the key scheduling algorithms is observed. The relationship is the same as was investigated in Chapter 5, where specifically, the analysis of the existence of a simple relationship in the linear diffusion transformations of SPN block ciphers was conducted. As the transformation is linear, it can be represented using a non-singular matrix. In the case of SMS4, in addition to linear components, the nonlinear components, which do not have equivalent matrix representations are also examined. For all components, the relationship is analyzed at the bit level\textsuperscript{2} because the size of the input word for these components is only 32 bits. Therefore, an exhaustive search over all input words is possible.

The linear relationship investigated in Chapter 5 is restated here, by considering a generic component $F$, which is an $n_b$-bit to $n_b$-bit function. For a component $F$, there exists a set of input words of $F$ which are equivalent to a simple rotation of the output word. That is, for some $n_b$-bit words $X_i$,

$$F(X_i) = X_i \ll j$$ (6.1)

for some particular rotation values of $j \in \{0, 1, \ldots, n_b - 1\}$. In this chapter, for the specific case of SMS4, the focus is on $n_b = 32$.

In the remainder of this section, $\hat{\mathcal{F}}_F$ denotes the total number of distinct values $X_i$ that satisfy the relationship described in Equation 6.1 for a particular component $F$. The set containing these input words $X_i$ is denoted by $\Theta_F$. Additionally, $\hat{\mathcal{F}}_{F,j}$ denotes the number of individual values that satisfy this relationship for a specific rotation value $j$. Note that the sum $\sum_{j=0}^{31} \hat{\mathcal{F}}_{F,j}$ may be higher than $\hat{\mathcal{F}}_F$ because some input words satisfy this relationship for multiple values of $j$. For instance, if $F(02020202) = 08080808$, then Equation 6.1 holds for $j = 2, 10, 18, 26$, which means that the input word 02020202 is counted four times.

\textsuperscript{2}In Chapter 5, the relationship is analyzed at the word level for the block ciphers AES and ARIA, and at the bit level for the block ciphers PRESENT and Serpent.
Chapter 6. Linearity within the SMS4 Block Cipher

Table 6.1: Values of $X_i$ (in the set $\Theta_S$) and $j$ such that $S(X_i) = X_i \ll j$.

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>$j$</th>
<th>$X_i$</th>
<th>$j$</th>
<th>$X_i$</th>
<th>$j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0A0A0A0A</td>
<td>1, 9, 17, 25</td>
<td>21210A0A</td>
<td>1</td>
<td>ABB4ABDE</td>
<td>16</td>
</tr>
<tr>
<td>0A0A0A21</td>
<td>1</td>
<td>21210A21</td>
<td>1</td>
<td>ABDEABB4</td>
<td>16</td>
</tr>
<tr>
<td>0A0A210A</td>
<td>1</td>
<td>2121210A</td>
<td>1</td>
<td>B4ABDEAB</td>
<td>16</td>
</tr>
<tr>
<td>0A0A2121</td>
<td>1</td>
<td>21212121</td>
<td>1, 9, 17, 25</td>
<td>B4B4DEDE</td>
<td>16</td>
</tr>
<tr>
<td>0A210A0A</td>
<td>1</td>
<td>245C245C</td>
<td>2, 18</td>
<td>B4DEB4DE</td>
<td>8, 24</td>
</tr>
<tr>
<td>0A210A21</td>
<td>1, 17</td>
<td>245C2626</td>
<td>2</td>
<td>B4DEDEB4</td>
<td>16</td>
</tr>
<tr>
<td>0A21210A</td>
<td>1</td>
<td>26245C26</td>
<td>2</td>
<td>D056D056</td>
<td>5, 21</td>
</tr>
<tr>
<td>0A212121</td>
<td>1</td>
<td>2626245C</td>
<td>2</td>
<td>DEABB4AB</td>
<td>16</td>
</tr>
<tr>
<td>0B0B0B0B</td>
<td>6, 14, 22, 30</td>
<td>26262626</td>
<td>2, 10, 18, 26</td>
<td>DEB4B4DE</td>
<td>16</td>
</tr>
<tr>
<td>210A0A0A</td>
<td>1</td>
<td>56D056D0</td>
<td>5, 21</td>
<td>DEB4DEB4</td>
<td>8, 24</td>
</tr>
<tr>
<td>210A0A21</td>
<td>1</td>
<td>5C245C24</td>
<td>2, 18</td>
<td>DEDEB4B4</td>
<td>16</td>
</tr>
<tr>
<td>210A210A</td>
<td>1, 17</td>
<td>5C262624</td>
<td>2</td>
<td>E7E7E7E7</td>
<td>4, 12, 20, 28</td>
</tr>
<tr>
<td>210A2121</td>
<td>1</td>
<td>ABABABAB</td>
<td>0, 8, 16, 24</td>
<td>FAFAFADA</td>
<td>5, 13, 21, 29</td>
</tr>
</tbody>
</table>

Nonlinear Transformation $S$

Recall that the nonlinear transformation $S$ consists of the application of a single $8 \times 8$ S-box $s$, applied four times in parallel. By reverse engineering, Liu et al. deduce that the S-box for SMS4 is constructed using an inversion in the finite field $\mathbb{F}_{25}$, which is similar to that of the AES. Note that the design of the S-box for the AES explicitly avoids fixed points $[62]$. However, one fixed point in $s$ was identified in this work. The fixed point is the 8-bit value AB (in hexadecimal). Thus, the nonlinear transformation $S$ also has a fixed point, which is the hexadecimal value ABABABAB.

In addition to this fixed point, there also exist other input words $X_i$ that satisfy the relationship $S(X_i) = X_i \ll j$ for some $j > 0$. For these particular input words, the transformation $S$ is basically linear. There are $\hat{F}_S = 39$ (including the fixed point) distinct input words $X_i$ that have a relationship of this form. Let $\Theta_S$ denote the set containing the exact values of these $X_i$, which are given in Table 6.1. The number, $\hat{F}_{S,j}$, of values that satisfy this relationship for $S$, for each rotation value $j$ is given in Table 6.2.

Linear Transformation $L$

Four fixed points ($j = 0$) and 1020 other ($j > 0$) input words $X_i$ that satisfy the relationship $L(X_i) = X_i \ll j$, i.e. $\hat{F}_L = 1024$ were identified. Let $\Theta_L$ denote the set containing the exact values of these $X_i$. For these input words, the linear
6.1. Observations on Components in the Round Functions

Table 6.2: Number of output words which are equivalent to the rotation of the input word by $j$ bits to the left ($0 \leq j \leq 31$), for each component function

<table>
<thead>
<tr>
<th>$j$</th>
<th>$\mathcal{F}_{S,j}$</th>
<th>$\mathcal{F}_{L,j}$</th>
<th>$\mathcal{F}_{T,j}$</th>
<th>$\mathcal{F}_{L'},j$</th>
<th>$\mathcal{F}_{T'},j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>11</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1024</td>
<td>8</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>256</td>
<td>1</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>16</td>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

transformation $L$ provides poor diffusion because the input bits of these words are not well scattered by $L$ when producing the output words. The number, $\mathcal{F}_{L,j}$, of values that satisfy this relationship for $L$, for each rotation value $j$ is given in Table 6.2.

Function $T$

As a nonlinear cryptographic component, the function $T$ of SMS4 should behave like a random permutation. As explained in Section 5.1.1, the probability that a random permutation has a certain number of fixed points is known from the literature.

By exhaustive search, 11 fixed points were found in the function $T$ of SMS4, i.e. values $X_i$ such that $T(X_i) = X_i$ (for $j = 0$). The fixed points are 0B0B0B0B, 3E973E97, 3AE2C6AD, 62D367B9, 973E973E, E2C6AD3A, D367B962, C6AD3AE2, 67B962D3, AD3AE2C6 and B962D367. For a random permutation, the probability of having 11 fixed points is approximately $p_{n,11} = 1/(11! \cdot e) \approx 9.216E - 9$, which is quite small. Interestingly, if the S-box of SMS4 is replaced by the S-box of the AES, there are no fixed points in the resulting function $T$.

---

3Refer to Section 5.1.1 for the exact formula.
Table 6.3: Values of $X_i$ (in the set $\Theta_T$) and $j$ such that $T(X_i) = X_i \ll j$.

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>$j$</th>
<th>$X_i$</th>
<th>$j$</th>
<th>$X_i$</th>
<th>$j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>02740274</td>
<td>2, 18</td>
<td>4F13E4B4</td>
<td>2</td>
<td>BB06C4A3</td>
<td>26</td>
</tr>
<tr>
<td>039A039A</td>
<td>1, 17</td>
<td>58434DF7</td>
<td>26</td>
<td>BE6CBE6C</td>
<td>15, 31</td>
</tr>
<tr>
<td>06C4A3BB</td>
<td>26</td>
<td>5CDE9B16</td>
<td>14</td>
<td>C4A3BB06</td>
<td>26</td>
</tr>
<tr>
<td>0A0A0A0A</td>
<td>3, 11, 19, 27</td>
<td>62D367B9</td>
<td>0</td>
<td>C6AD3AE2</td>
<td>0</td>
</tr>
<tr>
<td>0B0B0B0B</td>
<td>0, 8, 16, 24</td>
<td>679B62D3</td>
<td>0</td>
<td>C7E7C7E7</td>
<td>13, 29</td>
</tr>
<tr>
<td>1079D3A1</td>
<td>31</td>
<td>6CBE6CBE</td>
<td>15, 31</td>
<td>D367B962</td>
<td>0</td>
</tr>
<tr>
<td>13E4B44F</td>
<td>2</td>
<td>74027402</td>
<td>2, 18</td>
<td>D3A11079</td>
<td>31</td>
</tr>
<tr>
<td>165CDE9B</td>
<td>14</td>
<td>79D3A110</td>
<td>31</td>
<td>DE9B165C</td>
<td>14</td>
</tr>
<tr>
<td>16AF4D4B</td>
<td>15</td>
<td>973E973E</td>
<td>0, 16</td>
<td>E0E1F7E3</td>
<td>9</td>
</tr>
<tr>
<td>1A2A1A2A</td>
<td>1, 17</td>
<td>9A039A03</td>
<td>1, 17</td>
<td>E1F7E3E0</td>
<td>9</td>
</tr>
<tr>
<td>21212121</td>
<td>3, 11, 19, 27</td>
<td>9B165CDE</td>
<td>14</td>
<td>E2C6AD3A</td>
<td>0</td>
</tr>
<tr>
<td>22E59CB6</td>
<td>31</td>
<td>9C6B22E5</td>
<td>31</td>
<td>E3E0E1F7</td>
<td>9</td>
</tr>
<tr>
<td>26262626</td>
<td>4, 12, 20, 28</td>
<td>A11079D3</td>
<td>31</td>
<td>E4B44F13</td>
<td>2</td>
</tr>
<tr>
<td>2A1A2A1A</td>
<td>1, 17</td>
<td>A3BB06C4</td>
<td>26</td>
<td>E59CB622</td>
<td>31</td>
</tr>
<tr>
<td>3AE2C65AD</td>
<td>0</td>
<td>ABABABAB</td>
<td>2, 10, 18, 26</td>
<td>E7E7E7C7</td>
<td>13, 29</td>
</tr>
<tr>
<td>3E973E97</td>
<td>0, 16</td>
<td>AD3AE2C6</td>
<td>0</td>
<td>E7E7E7E7</td>
<td>6, 14, 22, 30</td>
</tr>
<tr>
<td>434DF758</td>
<td>26</td>
<td>AF4D4B16</td>
<td>15</td>
<td>F758434D</td>
<td>26</td>
</tr>
<tr>
<td>4B16AF4D</td>
<td>15</td>
<td>B44F13E4</td>
<td>2</td>
<td>F7E3E0E1</td>
<td>9</td>
</tr>
<tr>
<td>4D4B16AF</td>
<td>15</td>
<td>B622E59C</td>
<td>31</td>
<td>FAFAFAPA</td>
<td>7, 15, 23, 31</td>
</tr>
<tr>
<td>4DF75843</td>
<td>26</td>
<td>B962D367</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Similarly, there exist input words $X_i$ that satisfy the relationship $T(X_i) = X_i \ll j$ for $j > 0$. In total, there are $\hat{F}_T = 59$ distinct input words $X_i$ (including the fixed points) that satisfy this relationship. Let $\Theta_T$ denote the set containing the exact values of these $X_i$, which are given in Table 6.3. The number $\hat{F}_{T,j}$ for each value of $j$ is given in Table 6.2.

Recall that the function $T$ is composed of $S$ and $L$, i.e. $T = L \circ S$. The 39 input words contained in the set $\Theta_S$ do not all appear in the set $\Theta_T$. However, there are seven input words that appear in the intersection of the two sets, $\Theta_S \cap \Theta_T$. These input words are 0A0A0A0A, 0B0B0B0B, 21212121, 26262626, ABABABAB, E7E7E7E7 and FAFAFAPA.

**Linear Transformation $L'$**

By exhaustive search, no fixed points were found for $L'$. However, $\hat{F}_L' = 8$ distinct input words $X_i$ that satisfy the relationship $L'(X_i) = X_i \ll j$ for some $j > 0$ were identified. Let $\Theta_{L'}$ denote the set containing the exact values of these $X_i$. As a linear transformation, the diffusion provided by $L'$ is poor for these input words. Note that the size of the set $\Theta_{L'}$ is smaller than the size of $\Theta_L$, despite
6.1. Observations on Components in the Round Functions

Table 6.4: Values of $X_i$ (in the set $\Theta_T'$) and $j$ such that $T'(X_i) = X_i \ll j$.

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>$j$</th>
<th>$X_i$</th>
<th>$j$</th>
<th>$X_i$</th>
<th>$j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>02020202</td>
<td>4, 12, 20, 28</td>
<td>5228B69C</td>
<td>6</td>
<td>A66BA66B</td>
<td>10, 26</td>
</tr>
<tr>
<td>06C206C2</td>
<td>1, 17</td>
<td>52505250</td>
<td>4, 20</td>
<td>AAA027D5</td>
<td>23</td>
</tr>
<tr>
<td>087BO87B</td>
<td>4, 20</td>
<td>5522DB49</td>
<td>27</td>
<td>B0B0B0B0</td>
<td>3, 11, 19, 27</td>
</tr>
<tr>
<td>10B78569</td>
<td>2</td>
<td>58F758F7</td>
<td>8, 24</td>
<td>B69C5228</td>
<td>6</td>
</tr>
<tr>
<td>121121212</td>
<td>6, 14, 22, 30</td>
<td>5A5A5A5A</td>
<td>2, 10, 18, 26</td>
<td>B7856910</td>
<td>2</td>
</tr>
<tr>
<td>12161216</td>
<td>7, 23</td>
<td>61F161F1</td>
<td>14, 30</td>
<td>BBBBBBBB</td>
<td>2, 10, 18, 26</td>
</tr>
<tr>
<td>16121612</td>
<td>7, 23</td>
<td>64C164C1</td>
<td>9, 25</td>
<td>BAC74FDD</td>
<td>27</td>
</tr>
<tr>
<td>1B341B34</td>
<td>5, 21</td>
<td>6910B785</td>
<td>2</td>
<td>C16C164</td>
<td>9, 25</td>
</tr>
<tr>
<td>1D411D41</td>
<td>2, 18</td>
<td>6B66B66A</td>
<td>10, 26</td>
<td>C206C206</td>
<td>1, 17</td>
</tr>
<tr>
<td>22DB4955</td>
<td>27</td>
<td>74747474</td>
<td>3, 11, 19, 27</td>
<td>C74FDBBA</td>
<td>27</td>
</tr>
<tr>
<td>25A498A2</td>
<td>1</td>
<td>7B087B08</td>
<td>4, 20</td>
<td>CB1CBA1</td>
<td>14, 30</td>
</tr>
<tr>
<td>27D5AA0</td>
<td>23</td>
<td>856910B7</td>
<td>2</td>
<td>D5AA0A027</td>
<td>23</td>
</tr>
<tr>
<td>28B69C52</td>
<td>6</td>
<td>94949494</td>
<td>6, 14, 22, 30</td>
<td>D69AD69A</td>
<td>12, 28</td>
</tr>
<tr>
<td>32232232</td>
<td>3, 11, 19, 27</td>
<td>98A225A4</td>
<td>1</td>
<td>DB495522</td>
<td>27</td>
</tr>
<tr>
<td>341B341B</td>
<td>5, 21</td>
<td>9A69A69A</td>
<td>12, 28</td>
<td>DDBAC74F</td>
<td>27</td>
</tr>
<tr>
<td>411D411D</td>
<td>2, 18</td>
<td>9C5228B6</td>
<td>6</td>
<td>DFDDFDFD</td>
<td>3, 11, 19, 27</td>
</tr>
<tr>
<td>495522DB</td>
<td>27</td>
<td>A027D5AA</td>
<td>23</td>
<td>E1E1E1E1</td>
<td>7, 15, 23, 31</td>
</tr>
<tr>
<td>4F4F4F4F</td>
<td>5, 13, 21, 29</td>
<td>A1CBA1CB</td>
<td>14, 30</td>
<td>F161F161</td>
<td>14, 30</td>
</tr>
<tr>
<td>4FDBAC7</td>
<td>27</td>
<td>A225A498</td>
<td>1</td>
<td>F758F758</td>
<td>8, 24</td>
</tr>
<tr>
<td>50525052</td>
<td>4, 20</td>
<td>A498A225</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the fact that $L'$ has fewer rotations than $L$. The number $\hat{F}_{L',j}$ of values for each rotation value $j$ is given in Table 6.2.

Function $T'$

Unlike the function $T$, the function $T'$ has no fixed points. However, there still exist some input words $X_i$ that satisfy the relationship $T'(X_i) = X_i \ll j$ for some $j > 0$. In total, there are $\hat{F}_{T'} = 59$ distinct input words $X_i$ that satisfy this relationship. Let $\Theta_{T'}$ denote the set containing the exact values of these $X_i$, which are given in Table 6.4. The number $\hat{F}_{T',j}$ for each value of $j$ is given in Table 6.2.

Recall that the function $T'$ is composed of $S$ and $L'$, i.e. $T' = L' \circ S$. The number $\hat{F}_{T'}$ of input words in the set $\Theta_{T'}$ is about 7 times more than the same number for $\Theta_{L'}$, and 20 more than $\Theta_S$. Unlike the function $T$, the input words contained in the set $\Theta_S$ do not appear at all in the set $\Theta_{T'}$, i.e. $\Theta_S \cap \Theta_{T'} = \emptyset$. However, there exists a set of input words for which the functions $T$ and $T'$ produce the same output words. This relationship is discussed in the following section.
Chapter 6. Linearity within the SMS4 Block Cipher

6.1.2 Relationship between $T$ and $T'$

As noted in Section 2.4.6, the encryption and the key scheduling algorithms are nearly identical, differing only in the linear transformation. Eight input words were identified for which the transformation $L$ and $L'$ produce the same output words, i.e. $L(Y_i) = L'(Y_i)$. These input words $Y_i$ are 00000000, 33333333, 55555555, 66666666, 99999999, AAAAAAAA, CCCCCCCC and FFFFFFFF.

Recall that the nonlinear transformation $S$ is the same in both the functions $T$ and $T'$. If there exist some input words $Y_i$ such that $L(Y_i) = L'(Y_i)$, then there exist words $X_i = S^{-1}(Y_i)$ such that $T(X_i) = L(S(X_i)) = L'(S(X_i)) = T'(X_i)$. The eight input words $X_i$ are 71717171, 28282828, 97979797, A5A5A5A5, 1F1F1F1F, 18181818, 04040404 and B9B9B9B9.

6.1.3 On the Branch Number of $L'$

As noted in Chapter 5, a commonly used measure of diffusion for SPN block ciphers is the branch number [62]. Recall from Section 5.2 that for an SPN cipher, this number denotes the minimum number of active S-boxes for any two consecutive rounds. In the context of a generic Feistel cipher such as SMS4, this is not always true. However, the number can still be useful in cryptanalysis, particularly in differential cryptanalysis. Therefore, the branch number of a linear transformation can be defined as the minimum number of non-zero subword differences for any input and output pair of the transformation.
### Table 6.5: The input-output pattern distribution of \( L' \)

<table>
<thead>
<tr>
<th>( \Gamma_X )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>3</th>
<th>5</th>
<th>6</th>
<th>9</th>
<th>A</th>
<th>C</th>
<th>7</th>
<th>B</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>31</td>
<td>.</td>
<td>220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>3</td>
<td>31</td>
<td>.</td>
<td>1</td>
<td>220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>31</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>15</td>
<td>236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>.</td>
<td>242</td>
<td>210</td>
<td>220</td>
<td>252</td>
<td>n_{22}</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>218</td>
<td>250</td>
<td>218</td>
<td>250</td>
<td>n_{21}</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>210</td>
<td>220</td>
<td>252</td>
<td>242</td>
<td>n_{22}</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>.</td>
<td>7</td>
<td>1</td>
<td>3</td>
<td>380</td>
<td>370</td>
<td>338</td>
<td>236</td>
<td>n_0</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>251</td>
<td>218</td>
<td>250</td>
<td>235</td>
<td>n_{19}</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>222</td>
<td>252</td>
<td>242</td>
<td>228</td>
<td>n_{20}</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>240</td>
<td>248</td>
<td>242</td>
<td>249</td>
<td>249</td>
<td>251</td>
<td>n_{20}</td>
<td>n_{18}</td>
<td>n_{16}</td>
<td>n_{11}</td>
<td>n_{23}</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>245</td>
<td>252</td>
<td>254</td>
<td>242</td>
<td>249</td>
<td>250</td>
<td>n_1</td>
<td>n_5</td>
<td>n_4</td>
<td>n_{12}</td>
<td>n_{29}</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>253</td>
<td>249</td>
<td>252</td>
<td>245</td>
<td>252</td>
<td>242</td>
<td>n_3</td>
<td>n_2</td>
<td>n_5</td>
<td>n_{13}</td>
<td>n_{30}</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>1</td>
<td>3</td>
<td>.</td>
<td>250</td>
<td>250</td>
<td>243</td>
<td>253</td>
<td>249</td>
<td>243</td>
<td>n_{17}</td>
<td>n_{14}</td>
<td>n_9</td>
<td>n_{15}</td>
<td>n_{24}</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>253</td>
<td>251</td>
<td>250</td>
<td>252</td>
<td>n_8</td>
<td>n_6</td>
<td>n_8</td>
<td>n_7</td>
<td>n_6</td>
<td>n_{10}</td>
<td>n_{28}</td>
<td>n_{27}</td>
<td>n_{26}</td>
<td>n_{25}</td>
<td>n_{31}</td>
<td></td>
</tr>
</tbody>
</table>
The branch number is calculated using Equation 5.5, which is given in Section 5.2. The notations regarding branch numbers used in that section are used here. For SMS4, the input word to both \( L \) and \( L' \) is partitioned into \( m = 4 \) subwords. Therefore, the optimal branch number for both \( L \) and \( L' \) is 5. Zhang et al. [185] showed that the branch number of \( L \) is indeed optimal, and noted that the number of rotations and XOR operations used in \( L \) are the minimum needed to reach this optimal branch number. However, they did not investigate the branch number for \( L' \). In this work, the branch number for \( L' \) is determined using a computer program and by observing the input-output pattern distribution table defined as follows.

Let both \( \Gamma_{X_i} \) and \( \Gamma_{Y_i} \) denote binary vectors of length \( m = 4 \). Furthermore, let \( W[\Gamma_{X_i}][\Gamma_{Y_i}] \) denote the \( \Gamma_{X_i} \)-th row and \( \Gamma_{Y_i} \)-th column entry for the input-output pattern distribution table. The entries for this table are computed as follows. Initialize the counter \( W \) to all-zero. For every input \( X_i = 0, 1, \ldots, 2^{32} - 1 \), calculate the output \( Y_i = L'(X_i) \) and increment the counter \( W[\Gamma_{X_i}][\Gamma_{Y_i}] \). The resulting table for \( L' \) is given by Table 6.5 where the entry ‘.’ denotes zero, for simplicity. Due to size constraints, some values are denoted by \( n_i \) given as follows.

\[
\begin{align*}
\ n_0 &= 63688, \quad n_7 = 64023, \quad n_{14} = 64049, \quad n_{21} = 64082, \quad n_{28} = 16323877, \\
\ n_1 &= 63894, \quad n_8 = 64024, \quad n_{15} = 64050, \quad n_{22} = 64088, \quad n_{29} = 16324086, \\
\ n_2 &= 63895, \quad n_9 = 64025, \quad n_{16} = 64051, \quad n_{23} = 16323681, \quad n_{30} = 16324087, \\
\ n_3 &= 63919, \quad n_{10} = 64026, \quad n_{17} = 64057, \quad n_{24} = 16323702, \quad n_{31} = 4229286763, \\
\ n_4 &= 63930, \quad n_{11} = 64027, \quad n_{18} = 64061, \quad n_{25} = 16323764, \\
\ n_5 &= 63939, \quad n_{12} = 64032, \quad n_{19} = 64065, \quad n_{26} = 16323875, \\
\ n_6 &= 64019, \quad n_{13} = 64040, \quad n_{20} = 64070, \quad n_{27} = 16323876, \\
\end{align*}
\]

The branch number of \( L' \) can be determined by first searching in Table 6.5 for a non-zero entry \( W[\Gamma_{X_i}][\Gamma_{Y_i}] \) with \( \Gamma_{X_i} \neq 0 \) for which the sum of the Hamming weight for \( \Gamma_{X_i} \) and \( \Gamma_{Y_i} \) is the least among other entries. Then, the branch number is calculated as \( B(L') = wt(\Gamma_{X_i}) + wt(\Gamma_{Y_i}) \). An example of such an entry is \( W[1][7] \) and thus, the branch number of \( L' \) is \( B(L') = wt(1) + wt(7) = 1 + 3 = 4 \), which is not optimal.

The input-output pattern distribution table also gives information regarding possible and impossible subword difference paths propagated by \( L' \). This is useful for differential-type attacks. The sub-optimal branch number for \( L' \) is an indication of a potential weakness. This is exploited in Section 6.3 in a differential attack on a slightly modified variant of SMS4.
6.2 Cryptographic Significance

This section discusses the cryptographic significance of the observations made in Section 6.1.

6.2.1 Implications for the Key Scheduling Algorithm

The length of the master key for SMS4 is 128 bits, hence there are \(2^{128}\) possible values of the master key. The key scheduling algorithm produces 32 subkeys, each of 32 bits, thus the sequence of subkeys forms a \(32 \times 32 = 1024\)-bit binary sequence. Clearly, there are extremely many sequences of subkeys that are impossible.

Note that the function \(T'\), which is a 32-bit to 32-bit map, is bijective (using the theorem provided by Zhang et al. [185]). In every round, the value of a single 32-bit word is updated using the output of \(T'\), a function which takes the other three 32-bit words as input. After four rounds, all 128 bits of the master key are completely updated by the round functions. Therefore, it is reasonable to conjecture that all possible values of the first four subkeys are equally likely to occur (statistically independent), whereas the values for the remaining 28 subkeys are determined entirely by these four subkeys. This conjecture allows the following claim.

From Section 6.1.1, it is known that there are 59 distinct words \(X_i\) contained in the set \(\Theta_{T'}\). Recall that the value of the master key after the initialization phase is partitioned into four 32-bit words \((K_{-4}, K_{-3}, K_{-2}, K_{-1})\) and the \(i\)-th round constant is denoted by \(CK_i\). If the input words to the first four consecutive functions \(T'\) of the key scheduling algorithm are in the set \(\Theta_{T'}\), then the first four subkeys consist of merely linear combinations of the master key\(^4\). This event is illustrated as follows. If \((K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \in \Theta_{T'}\), then

\[
K_0 = K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0].
\]

\(^4\)Note that the initialization phase does not have any cryptographic significance. Therefore, if the value of the resulting key after this phase is known, then the value of the master key is also known.
Similarly, if \((K_{-2} \oplus K_{-1} \oplus K_0 \oplus CK_1) \in \Theta_T\), then

\[
K_1 = K_{-3} \oplus [(K_{-2} \oplus K_{-1} \oplus K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus CK_1) \ll j_1].
\]

Furthermore, if \((K_{-1} \oplus K_0 \oplus K_1 \oplus CK_2) \in \Theta_T\), then

\[
K_2 = K_{-2} \oplus [(K_{-1} \oplus K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus
K_{-3} \oplus [(K_{-2} \oplus K_{-1} \oplus K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus
CK_1) \ll j_1] \oplus CK_2) \ll j_2].
\]

Finally, if \((K_0 \oplus K_1 \oplus K_2 \oplus CK_3) \in \Theta_T\), then

\[
K_3 = K_{-1} \oplus [(K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus
K_{-3} \oplus [(K_{-2} \oplus K_{-1} \oplus K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus
CK_1) \ll j_1] \oplus
K_{-2} \oplus [(K_{-1} \oplus K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus
K_{-3} \oplus [(K_{-2} \oplus K_{-1} \oplus K_{-4} \oplus [(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0) \ll j_0] \oplus
CK_1) \ll j_1] \oplus CK_2) \ll j_2] \oplus CK_3) \ll j_3].
\]

The above linear equations are valid for specific values of \(j_i \in \{0, 1, \ldots, 31\}\). This event occurs with probability \((59/2^{32})^4 \approx 2^{-104.5}\) and thus, there are approximately \(2^{3.5}\) values of the master key which cause such an event to happen.

### 6.2.2 Implications for the Encryption Algorithm

As noted in Section 6.1.1, there are 59 distinct words \(X_i\) contained in the set \(\Theta_T\). If the input words to the first four consecutive functions \(T\) of the encryption algorithm are in the set \(\Theta_T\), then the output block after four rounds consist of merely linear combinations of the plaintext block and subkeys. In general, this event is similar to that described in Section 6.2.1. A specific case in which only fixed points occur in the first four consecutive rounds will now be demonstrated. Let \(\hat{\Theta}_T\) denote a subset of \(\Theta_T\) containing the 11 fixed points for \(T\) (Refer to Section 6.1.1). This event is shown as follows for the plaintext block \(P = (X_0, X_1, X_2, X_3)\) and subkeys \(K_0, K_1, K_2\) and \(K_3\). If
\[(X_1 \oplus X_2 \oplus X_3 \oplus K_0) \in \hat{\Theta}_T \text{ then} \]

\[X_4 = X_0 \oplus X_1 \oplus X_2 \oplus X_3 \oplus K_0.\]

Similarly, if \((X_2 \oplus X_3 \oplus X_4 \oplus K_1) \in \hat{\Theta}_T\), then

\[X_5 = X_0 \oplus K_0 \oplus K_1. \quad (6.2)\]

Furthermore, if \((X_3 \oplus X_4 \oplus X_5 \oplus K_2) \in \hat{\Theta}_T\), then

\[X_6 = X_1 \oplus K_1 \oplus K_2. \quad (6.3)\]

Finally, if \((X_4 \oplus X_5 \oplus X_6 \oplus K_3) \in \hat{\Theta}_T\), then

\[X_7 = X_2 \oplus K_2 \oplus K_3. \quad (6.4)\]

Clearly, for the specific case of fixed points, the linear relationships above are much simpler than the general case because some words \(X_i\) and subkeys \(K_i\) cancel. This specific event occurs with probability \((11/2^{32})^4 \approx 2^{-114.2}\) and thus, there are approximately \(2^{13.8}\) values of the plaintext block that cause such an event to happen for the full SMS4. In the general case, there are \(2^{23.5}\) values of the plaintext block that cause the four-round linearity to happen.

### 6.2.3 Further Implications for Both the Key Scheduling and the Encryption Algorithms

The points discussed in Sections 6.2.1 and 6.2.2 have further security implications for SMS4. In the (admittedly rare) event that both the key scheduling and the encryption algorithms behave linearly for the first four rounds, the output block after four rounds of SMS4 is composed of merely linear combinations of the plaintext block and subkeys. The subkeys, in turn, are composed of linear combinations of the master key. Theoretically, if both of these events occur at the same time, then the number of effective rounds for SMS4 is reduced by four, from 32 to 28.

The above discussions only consider the case for which the linearity occurs in the key scheduling and the encryption algorithms in the first four consecutive rounds. Note that it may be possible for the linearity to occur in any four
of the 32 rounds of SMS4. Furthermore, for certain particular combinations of plaintext block and master key, the linearity might exist in more than four rounds. In order to determine the veracity of this claim, a comprehensive study of such combinations needs to be undertaken. In this case, the number of effective rounds for SMS4 can be further reduced.

6.2.4 Susceptibility to Algebraic Attack

The algebraic attack\cite{55} introduced by Courtois and Pieprzyk, which is explained in Section 2.3.9 consists of building a system of binary equations that link the plaintext, subkeys and ciphertext blocks. The equations are then solved to obtain the subkey bits.

The entire 32 rounds of SMS4 can be described by a system of binary equations. A subset of the binary equations describing the S-box of SMS4, which is based on a finite field inversion\cite{125}, are quadratic whereas the remaining equations are linear. In general, one of the obstacles in solving this kind of equation system is the existence of quadratic equations.

As discussed in Sections 6.2.1, 6.2.2 and 6.2.3, there exist a few exceptional cases in which the nonlinear functions $T$ and $T'$ are linear in the first four rounds of SMS4. Under these conditions, the binary equations describing the first four rounds are also entirely linear. Therefore, there is no need to describe the S-boxes in these rounds as systems of quadratic equations. Since the occurrence of this event is statistical in nature, more plaintext and ciphertext pairs may be needed compared to a conventional algebraic attack. However, the removal of some quadratic equations might help in reducing the complexity of solving the equation system.

6.2.5 Susceptibility to Advanced Variants of the Slide Attack

The slide attack, described in Section 2.3.7 is introduced by Biryukov and Wagner\cite{35,36}. Recall that if two different plaintexts are given, the attack permits the sliding of the two encryptions by a certain number of rounds. This is due to the similarity that exists between the structure of the two encryptions. The attack also allows the sliding of encryption with decryption\cite{36}.

It has been shown in Section 6.1.2 that there are eight input words for which
the functions $T$ and $T'$ produce the same output words. This similarity might provide an avenue for advanced variants of the slide attack. However, it is an open problem to determine whether it is useful to slide the encryption algorithm with the key scheduling algorithm if both algorithms are nearly identical, as is the case for SMS4.

6.2.6 Subkeys and Related-Keys

As discussed in Section 6.2.1, it was conjectured that all possible 32-bit subkey values of the first four rounds of SMS4 are equally likely to occur. This allows the exploration of the relationship between subkeys in these rounds and subkeys in subsequent rounds. One possible relationship is described as follows. If the first four 32-bit round subkeys are identical (that is $K_i = \hat{K}$ for $i = 0, 1, 2, 3$ where $\hat{K}$ denotes an arbitrary 32-bit value), then a total of $2^{32}$ (out of $2^{128}$) master keys have the following forms:

$$K_{-1} = \hat{K} \oplus T'(\hat{K} \oplus CK_3), \\
K_{-2} = \hat{K} \oplus T'(K_{-1} \oplus CK_2), \\
K_{-3} = \hat{K} \oplus T'(K_{-2} \oplus K_{-1} \oplus \hat{K} \oplus CK_1) \text{ and } K_{-4} = \hat{K} \oplus T'(K_{-3} \oplus K_{-2} \oplus K_{-1} \oplus CK_0).$$

If this event and the event discussed in Section 6.2.2 occur at the same time, then the subkeys that exist in Equations 6.2, 6.3, and 6.4 will cancel and the subkeys in the first four rounds will have no effect on the intermediate words $X_5$, $X_6$ and $X_7$.

Similarly, if the subkeys in the first four rounds are identical, then the subkeys in rounds four ($K_4$) and five ($K_5$) have the following form:

$$K_4 = \hat{K} \oplus T'(\hat{K} \oplus \hat{K} \oplus \hat{K} \oplus CK_4) \oplus T'(\hat{K} \oplus CK_4)$$

Suppose that $K_4 = \hat{K}$, which implies that $K_4 = \hat{K} = \hat{K} \oplus T'(\hat{K} \oplus CK_4)$ and $T'(\hat{K} \oplus CK_4) = 0$. Since $CK_4$ is a known fixed round constant, only one value of $\hat{K}$ can satisfy this equation, that is $\hat{K} = CK_4 \oplus 71717171 = 1060FF4$. Therefore, for all $2^{32}$ master keys that have the form $K_i = \hat{K}$ for $i \in \{0, 1, \ldots, 4\}$, only one master key satisfies the relationship $K_3 = K_4$. The remaining $2^{32} - 1$ master keys have the relationship $K_3 \neq K_4$. Stated differently, if a sequence of subkeys is given containing five identical words $K_i = \hat{K}$ for $i \in \{0, 1, \ldots, 4\}$, and $K_i \neq 1060FF4$, then it is known that the subkeys are not the first four subkeys derived from the SMS4 key scheduling algorithm. These kinds of relationships can be further investigated beyond the first four rounds by taking into consideration the relationship between the round constants. The algorithm to derive these
constants is already given in Section 2.4.6. In a key recovery attack, if the attacker knows the relationship of the words in the master key beforehand, then guesses that are impossible can be skipped. This reduces the key space that the attacker needs to guess.

The previous single-key discussion may possibly be extended to the related-key model, which is explained in Section 2.3.8. Related-key attacks \[11, 106\] allow the attacker to choose the relationship between two different master keys but not the actual value of the keys. The relationship is chosen such that the round subkeys of the first master key are related in some way to the round subkeys of the second master key. Then, several (known or chosen) plaintexts are encrypted using these related master keys to obtain the corresponding ciphertexts. The ciphertexts are then used to recover both master keys. This is an area for further investigation.

### 6.3 A Differential Attack on Modified SMS4

This section presents a differential attack \[24\], which is explained in Section 2.3.2, on a modified variant of SMS4, created by replacing the linear transformation \(L\) in the encryption algorithm with \(L'\). This basically means that the attack is on the key scheduling algorithm, as if it was used for encryption. It will be demonstrated next that a differential attack is possible on a 27-round version of this variant.

#### 6.3.1 23-Round Characteristic

A 5-round self-iterating differential characteristic based on previous differential attacks on SMS4 \[104,183,185\] is used. The characteristics used in these attacks have six active S-boxes: three in the fourth round and three in the fifth. Based on the entries of the input-output pattern distribution of \(L'\) given in Table 6.5, it is known that there exists a number of differential paths where only two S-boxes are active in one round. An example of such a path is the entry \(W[3][3]\).

Let \(\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)\) denote a 32-bit difference formed from the concatenation of four 8-bit differences \(\alpha_i\). The 5-round self-iterating characteristic satisfies \(0 \overset{T}{\rightarrow} 0\) in the first, second and third rounds; and \(\alpha \overset{T}{\rightarrow} \alpha\) in the fourth and fifth rounds. This characteristic is given as follows: \((\alpha, \alpha, \alpha, 0) \rightarrow (\alpha, \alpha, 0, \alpha) \rightarrow (\alpha, 0, \alpha, \alpha) \rightarrow (0, \alpha, \alpha, \alpha) \rightarrow (\alpha, \alpha, \alpha, \alpha) \rightarrow (\alpha, \alpha, 0)\).
6.3. A Differential Attack on Modified SMS4

By exhaustive search, six values of $\alpha$ have been found that satisfy the above 5-round self-iterating characteristic such that only two bytes of $\alpha$ are nonzero (i.e., two bytes are active). The values are 0000900C, 00C900C9, 00900C00, 0C000090, 900C0000 and C900C900. The probability that $\alpha \xrightarrow{T} \alpha$ for each of these values is $2^{-14}$. The probability for the 5-round self-iterating characteristics is therefore $(2^{-14})^2 = 2^{-28}$. This characteristic can be concatenated four and a half times to produce a 23-round differential characteristic with total probability $(2^{-28})^4 = 2^{-112}$ given below.

\[
(\alpha, \alpha, \alpha, 0) \xrightarrow{5 \text{ Rounds}} (\alpha, \alpha, \alpha, 0) \xrightarrow{5 \text{ Rounds}} (\alpha, \alpha, \alpha, 0) \xrightarrow{3 \text{ Rounds}} (0, \alpha, \alpha, \alpha)
\]

In comparison, the best 5-round differential characteristic on the original SMS4 has probability $2^{-38}$ and can only be concatenated up to three and a half times (to construct an 18-round differential characteristic) with total probability $2^{-114}$ [185].

6.3.2 27-Round Key Recovery Attack

The previous 23-round differential characteristic can be used in a 27-round key recovery attack on the modified variant of SMS4. Since the attack is heavily based on previous differential attacks [104, 183, 185], only a brief description of the attack is presented.

Choose $\alpha = (00, 00, 90, 0C)$ and let $\Lambda$ be the set of all output differences of $T'$ where only 2 S-boxes are active. For each S-box, there are only 127 possible output differences. Therefore, the set contains $127 \cdot 2 \approx 2^8$ possible values.

Let $P$ and $P^*$ denote a plaintext pair and let $C$ and $C^*$ denote the corresponding ciphertext pair after 27 rounds, where $P = (X_0, X_1, X_2, X_3)$, $P^* = (X'_6, X'_1, X'_2, X'_3)$, $C = (X_{27}, X_{28}, X_{29}, X_{30})$ and $C^* = (X'_{27}, X'_{28}, X'_{29}, X'_{30})$. The attack proceeds as follows.

1. Generate $m \cdot (2^{16})^3 = m \cdot 2^{48}$ plaintext blocks where bytes 2, 3, 6, 7, 10 and 11 are set to all possible values whereas the remaining bytes are fixed. These plaintext blocks produce $m \cdot 2^{48}/2 = m \cdot 2^{47}$ plaintext pairs $(P, P^*)$ having the difference $(\alpha, \alpha, \alpha, 0)$.

2. Encrypt the plaintexts using 27 rounds of the modified SMS4.
3. Filter the ciphertexts so that only ciphertexts with \((X_{27} \oplus X_{27}^*) \in \Lambda\) are chosen. This filtering causes about \(m \cdot 2^{47} \cdot 2^{-8} = m \cdot 2^{39}\) pairs to remain.

4. Let \(\gamma_{i,j} = s(X_{i,j} \oplus X_{i+1,j} \oplus X_{i+2,j} \oplus K_{i-1,j}) \oplus s(X_{i,j}^* \oplus X_{i+1,j}^* \oplus X_{i+2,j}^* \oplus K_{i-1,j})\) and \(\delta_{i,j} = L'(X_{i+3,j} \oplus X_{i+3,j}^* \oplus \alpha_j)\).

5. For each round \(i = 27, 26, 25\), do the following

   (a) For each byte \(j = 0, 1, 2, 3\), do the following

      i. For each byte guess \(K_{i-1,j} = 0, 1, \ldots, \text{FF}\), do the following

         A. Calculate \(\gamma_{i,j}\) and \(\delta_{i,j}\).

         B. If \(\gamma_{i,j} = \delta_{i,j}\), then store \(K_{i-1,j}\) as a possible correct candidate key byte.

      ii. After all values have been guessed for this byte, wrong pairs are expected to be discarded by a factor of \(2^{-8}\).

6. After Step (5), 12 bytes of key material have been guessed and about \(m \cdot 2^{39} \cdot (2^{-8})^{12} = m \cdot 2^{-57}\) pairs are expected to remain.

7. For round \(i = 24\), do the following

   (a) For each byte guess \(K_{23,0} = 0, 1, \ldots, \text{FF}\), calculate \(\gamma_{24,0}\) and \(\delta_{24,0}\). If \(\gamma_{24,0} = \delta_{24,0}\), then store \(K_{23,0}\) as a possible correct candidate key byte.

   (b) After all values have been guessed for this byte, wrong pairs are expected to be discarded by a factor of \(2^{-8}\).

8. After Step (7), about \(m \cdot 2^{-57} \cdot (2^{-8}) = m \cdot 2^{-65}\) pairs are expected to remain. If \(m = 2^{68}\), then for a wrong key guess, the expected number of remaining ciphertext pairs is approximately \(2^{68} \cdot 2^{-65} = 2^3 = 8\). However, for a right key guess, the expected number of remaining ciphertext pairs is approximately \(2^{68} \cdot 2^{48} \cdot 2^{-112} = 2^4 = 16\).

9. If the guesses for \(K_{23,0}, K_{24}, K_{25}\) and \(K_{26}\) suggest more than 16 remaining ciphertext pairs, then the guesses are candidates for correct subkeys.

The data complexity of this 27-round attack is \(2^{68} \cdot 2^{48} = 2^{116}\) chosen plaintexts. The time complexity of the attack is dominated by Steps (5) and (7). At the beginning of Step (5), there are about \(2^{68} \cdot 2^{39}\) pairs of texts. A total of 12 bytes of key material are guessed and for each guess, wrong pairs are discarded
6.4 Summary and Conclusion

by a factor of $2^{-8}$. At the beginning of Step (7), there are roughly $2^{68} \cdot 2^{-57}$ pairs of texts and only one byte of key material is guessed. Adding these two complexities together, the total time complexity of the attack is approximately $\sum_{k=0}^{11} 2^8 \cdot 2^{68} \cdot 2^{39} \cdot 2^{-8k} + 2^8 \cdot 2^{68} \cdot 2^{-57} \approx 2^{115}$ encryptions. In contrast, the best existing cryptanalysis on the original SMS4 is a differential attack on 22 rounds with a data complexity of $2^{117}$ chosen plaintexts and time complexity of $2^{112.3}$ 22-round encryptions [185].

6.3.3 Comments on the Security of SMS4

As mentioned at the beginning of Section 6.3, the attack described above is the same as attacking the key scheduling algorithm, as if it was used for encryption. The original components of SMS4 are used and no modification of the components was done. The key scheduling algorithm might therefore be exploited in related-key differential attacks.

In the light of the discussion in Section 6.2.3, there is a small possibility that the first four rounds of SMS4 are deprived of nonlinearity. Under these conditions, the number of effective rounds for SMS4 is theoretically reduced by four, from 32 to 28. In this section, an attack against 27 rounds of a slightly modified variant of SMS4 has been demonstrated. This is only one round short of the effective 28 rounds. Note that the four-round linearity event discussed in Section 6.2.3 refers to the event in which the function $T$ was used in the encryption, instead of $T'$, as is the case here. However, if $T'$ was used in the encryption, the probability of this event to occur for $T'$, in the general case, is the same as if $T$ was used in the encryption. This is because the number of input words in the set $\Theta_T$ is the same as the set $\Theta_{T'}$.

Recall that the best attack on the original SMS4 is on 22-rounds [185], which is six rounds short of the effective 28 rounds. However, note that the security margin is reduced from 32 to 28 rounds only if the linearity in the first four rounds can be detected and utilized in an attack. A method to detect this remains an open problem.

6.4 Summary and Conclusion

This chapter presents several new observations on both the encryption and the key scheduling algorithms of the SMS4 block cipher. The existence of fixed
points and of simple linear relationships between the bits of the input and output words for each component of the round functions for some input words has been shown. Furthermore, it has been determined that the branch number of the linear transformation in the key scheduling algorithm is less than optimal.

The major security implication of these observations is that, for all keys, the round function is not always nonlinear. Due to this linearity, for some combinations of plaintext block and master key, the number of effective rounds of SMS4 is theoretically reduced by four, from 32 to 28. A brief exploration of the susceptibility of SMS4 against algebraic and advanced variants of the slide attacks was conducted.

Finally, it has been demonstrated in this chapter that if the linear transformation $L$ of the encryption algorithm is replaced with the linear transformation $L'$ of the key scheduling algorithm, then this variant of SMS4 is weaker than the original SMS4 with regard to differential cryptanalysis. This is illustrated by attacking 27 rounds of this modified variant, compared to the best existing differential attack on 22 rounds of the original SMS4. This is possible due to the sub-optimal branch number of $L'$. This property of $L'$ might be an indication of further weakness that can be exploited in an attack. It is strongly believed that this variant is also weaker than SMS4 against other differential-type attacks.

Given the number of expected fixed points, it is unlikely that the components in the round functions are generated randomly, that is, they were selected specifically. However, the criteria for selecting the components are not known. The findings made in this chapter raise serious questions on the security provided by SMS4, and might provide clues on the existence of flaws in the design of the cipher.
Chapter 7

Conclusions and Open Problems

Linear components are commonly used in block ciphers to ensure that the ciphers are efficient. Linear components by themselves, however, are not enough to achieve a high level of security. Thus, nonlinear components are needed so that no linear relationship exists between the inputs and outputs of the cipher that can be exploited in attacks. These two components, combined with the unknown master key, provide the necessary tools to ensure that the relationship between the plaintext, ciphertext and master key blocks is highly nonlinear.

This thesis presents the results of an investigation into the linear relationships in block ciphers. Although the round function of a block cipher is supposed to act as a nonlinear function, some linearity can still exist if the plaintexts and the components of the cipher are manipulated in particular ways. Certain manipulation techniques have been investigated in this thesis, and the results of those investigations are discussed.

This chapter is organized as follows. Section 7.1 gives a review of the contributions of this thesis. Some open problems for future work are proposed in Section 7.2.

7.1 Review of Contributions

In this thesis, several linear relationships in block ciphers have been identified and used to improve existing attacks, to identify unexpected properties in nonlinear transformations, and as basis for a new measure of diffusion. A review of the
major contributions presented in each chapter is as follows.

7.1.1 Chapter 3

This chapter introduced a new attack called the bit-pattern based integral attack, which exploits linear relationships existing in the texts at intermediate stages of encryption. The attack provides new tools to enable the traditional word-based integral attacks [34, 59, 112, 129] to be applied to bit-based block ciphers. The approach differs from the traditional word-based integral attacks in some respects. Firstly, the bits of the texts throughout the bit-pattern based attack are treated as an ordered set. The notations to describe the attack are thus more refined than the traditional word-based integral attack. Secondly, a basic bit-pattern based integral distinguisher uses more than one active S-box in the first round. This method is used to facilitate the tracing of the bit patterns throughout the cipher. In contrast, the traditional integral distinguisher makes use of only a single active S-box.

The bit-pattern based integral attack was applied successfully, for the first time, to Noekeon reduced to five (out of 16) rounds, PRESENT reduced to seven (out of 31) rounds and Serpent reduced to six (out of 32) rounds. Piret [154, pg 79-82] had previously attempted to apply an integral attack to Serpent but the distinguisher used by Piret does not occur with probability one. Also, the number of attacked rounds was not mentioned. Additionally, it has been shown in this thesis that bit-pattern based integral attacks on these ciphers are much better in terms of data complexity than comparable existing differential attacks.

There is a general belief that integral attacks are not suitable for application to bit-based block ciphers. Note that the integral attack originated from the same family as the Square attack introduced in 1997 [59]. However, the block ciphers Serpent [4] and Noekeon [60], which were respectively proposed in 1998 and 2000, do not mention the applicability of the Square attack. Furthermore, the designers of PRESENT note that “the design of PRESENT is almost exclusively bitwise, and while the permutation operation is somewhat regular, the development and propagation of word-wise structures are disrupted by the bitwise operations used in the cipher” [37]. In this chapter, contrary to the general belief, it has been shown that integral attack can still be applied to reduced-round versions of bit-based block ciphers, including PRESENT.
7.1.2 Chapter 4

This chapter presented the analysis of the LEX-AES stream cipher in relation to algebraic attacks. The LEX-AES is an example of using the block cipher AES in the generic LEX technique. The analysis makes use of the linear and nonlinear relationships which are present between the bytes of the internal state and round subkey. It is believed that the results presented in this chapter are the first attempts at creating a solvable equation system for stream ciphers based on the LEX method. The results of this chapter include analysis on LEX-AES and LEX-BES.

For LEX-AES, it is shown that a novel system containing 21 equations and 17 byte variables (consisting of 4 state and 13 subkey variables) over $GF(2^8)$ can be constructed that describes the cipher. The security of LEX-AES therefore relies on the difficulty of solving this system of equations. Note that an exhaustive key search of LEX-AES includes trying all possible 16 bytes of the master key. The number of variables in the system of equations presented in this chapter is 17, and each variable is one byte. Thus, solving the system of equations is close to an exhaustive key search. Only 36 bytes of keystream are needed to generate the equations. This makes the attack more threatening in the real world compared to the current best attack on LEX-AES which needs almost 80 billion ($2^{36.3}$) bytes of known keystream [68]. Furthermore, previous algebraic attack attempts on the AES (which also applies to LEX-AES) make use of the inversion based equation, which is not valid for all values of the state and keystream bytes [55, 77, 136]. The system of equations presented in this chapter does not suffer from this problem. Note that the variant of LEX studied in this chapter is based on the AES block cipher, which is regarded as a strong block cipher. However, the security margin was found to be low. Hence, at the moment, LEX cannot be regarded as a secure generic method.

For LEX-BES, it was found that attempts at solving the resulting system of equations over $GF(2^8)$, even for a very small-scale variant, seem futile. The MAGMA 2.14-11 computational algebra package [38], failed to compute the Gröbner basis for the related polynomial ideal even for a system containing 120 equations and 76 variables over $GF(2^4)$. The results are similar to those obtained by Cid, Murphy and Robshaw in their investigation of small scale variants of the BES [46]. This indicates that the same technique would most likely fail for the full-scale variant, which involves 7168 equations in 3584 variables over $GF(2^8)$. 
7.1.3 Chapter 5

In this chapter, a new measure of diffusion provided by the linear transformation of SPN block ciphers is proposed. The proposed measure is based on the existence of fixed points and of very simple linear relationships linking the input and output bits of the linear transformation. The measure gives the average fraction of input words to a linear diffusion transformation that are left unchanged by the transformation. The measure is applied to the block ciphers AES, ARIA, PRESENT and Serpent. It is shown that except for Serpent, the linear transformations of the analyzed ciphers do not behave as expected for a random linear transformation.

It is also suggested that a connection exists between the number of fixed points and existing design criteria for the linear transformation, namely branch number and efficiency, although the exact relationship remains unclear. This was explored by investigating alternate matrix representations for the linear transformation. For instance, increasing the branch number may also increase the number of fixed points. This was the case for the alternate matrices for the AES and Serpent. However, matrices were found which had the same branch number, but reduced the number of fixed points for the AES, ARIA and PRESENT. For the AES, this additionally comes at the expense of slow decryption. No linear transformations were found that simultaneously have an optimal branch number, reasonably high performance and a number of fixed points within the expectation of a random linear transformation for any of the block ciphers examined. Additionally, possible techniques to exploit the existence of a large number of fixed points in cryptanalysis were explored. Although no major attack was found, it is believed that the number of fixed points is still an important property that should be considered in cipher design.

7.1.4 Chapter 6

This chapter presents the identification of several new and previously unknown simple linear relationships in the nonlinear transformations of the SMS4 \cite{64} \cite{149} block cipher. The chapter also explores potential weaknesses in the key scheduling algorithm of the cipher.

One of the findings made is the existence of 59 (out of $2^{32}$) distinct output words of both nonlinear transformations $T$ and $T'$ which are equivalent to
mere rotations of the input word. These two transformations provide the crucial nonlinearity component in every round of the encryption and key scheduling algorithms, respectively. In particular, for the transformation $T$, 11 of the identified 59 output words are fixed points. Therefore, the nonlinear transformation $T$ clearly does not appear random. Due to this linearity, there is some exceptional but surprising cases in which the first four rounds of SMS4 are linear. In these cases, the effective number of rounds for SMS4 is reduced by four, from 32 to 28. Note that it may be possible for the linearity to occur in any four of the 32 rounds of SMS4. Furthermore, for certain particular combinations of plaintext block and master key, the linearity might possibly exist in more than four rounds. In this case, the number of effective rounds for SMS4 can be further reduced.

In this chapter, it is also shown that the branch number for the linear transformation used in the key scheduling algorithm is not optimal. It is identified that the branch number for this transformation is four, which is one short of the optimal value. As a consequence, it is demonstrated that if this linear transformation is used in the encryption algorithm, then a 27-round (out of 32) version of the resulting modified variant of SMS4 is vulnerable to a differential attack. These findings are the first to show that the key scheduling algorithm is weaker than the encryption algorithm with respect to differential cryptanalysis.

As far as the author knows, the observations are novel and reveal some peculiarities inherent in the components of the SMS4 block cipher. Since the SMS4 block cipher is used in the Chinese Wireless LAN Wired Authentication and Privacy Infrastructure (WAPI), the security provided warrants thorough investigation.

7.2 Open Problems

In Chapter 3, the bit-pattern based integral attack has been applied to the block ciphers Noekeon, PRESENT and Serpent. One area of further research is to apply the attack to other bit-based block ciphers such as HIGHT [89] and the KATAN and KTANTAN [42] families. It is also interesting to apply the bit-pattern based integral attack to existing word-based block ciphers which are vulnerable to the word-based integral attacks. For instance, by using the method developed in this thesis, Wei et al. [176] manage to construct a 3-round bit-pattern based integral distinguisher for the 256-bit key AES. The distinguisher requires fewer
chosen plaintexts than the traditional word-based integral distinguisher. However, the bit-based integral distinguisher might not be easily extended if rounds are appended at the top of the distinguisher. This is because, in a typical word-based integral distinguisher, the number of active S-boxes in the first round is one. In the bit-based integral approach, there are several active S-boxes in the first round. Therefore, adding rounds at the top of the distinguisher requires more subkey guesses in the bit-based case, compared to the word-based case. Techniques to address this problem are another area of further investigation.

It might also be possible to combine algebraic techniques with the integral attack (word- and bit-based). In a basic integral distinguisher, the XOR sum of the texts in the intermediate stages of the first few rounds of encryption is zero. This relationship introduces linear equations to the system of equations describing the cipher. A similar work in this direction is the proposed usage of algebraic techniques in differential cryptanalysis [3]. However, this proposal deals with equations that are valid with some non-negligible probability. In comparison, the equations arising from the XOR sum of the texts are valid with probability one.

In Chapter 5, the LEX-AES was examined and it is discovered that the cipher can be described using a very small system of equations. If other block ciphers are used in this manner, then what are the security ramifications? This is an area of further investigation. Furthermore, one can also examine the security of LEX-AES if the number of keystream byte leakage are reduced, or if the keystream bytes are leaked at longer intervals.

It is also worthwhile investigating efficient methods for solving the system of equations arising from LEX-AES. As with other algebraic approaches, the lack of practical methods in finding the solutions of the equations have been the major stumbling block in launching successful algebraic attacks. Though several equation-solving methods have been proposed [52, 55, 105, 170], none so far manage to demonstrate a convincing key-recovery attack. The work in this direction is crucial because algebraic techniques typically require only a very small amount of known plaintext. Therefore, further investigation needs to be done in this area.

In Chapter 5, the relationship between fixed points and existing design criteria for linear diffusion transformations of SPN block ciphers was analyzed. This was conducted by investigating different matrix representations of the linear
transformations of several block ciphers. However, not all possible matrices were searched. Therefore, a more comprehensive study should be done in order to determine the relationship. There is also a need to further investigate whether the existence of a large number of fixed points in a linear transformation can be exploited in cryptanalysis. Although some methods of exploiting the fixed points have been proposed in this chapter, none yield attacks which perform better than existing attacks.

In Chapter 6, it was shown that in some rare cases, the first four rounds of SMS4 are linear. The existence of this property might be combined with existing cryptanalytic techniques in order to construct new distinguishers for SMS4 or improve current attacks. One possible way is to use the algebraic-differential technique, described earlier, to launch attacks exploiting this linearity.

The thesis also explores the round subkey sequences of the SMS4 block cipher. If it is feasible to construct classes of possible or impossible subkey sequences, then in a key recovery attack, one might be able to skip subkey values that are known to be impossible. This helps reduce the number of guesses and improves the time complexity of a key recovery attack. Techniques to manipulate this property in the related-key model are also areas of further investigation.
Appendix A

S-Boxes

The tables presented in this appendix represent the S-boxes used by each of the block ciphers defined in Section 2.3. Note that both S-boxes s₁ of ARIA and AES use the same mapping. Furthermore, note that the mapping for the S-box of Noekleon presented here is slightly different compared to that given in the specification [60]. This is because in this thesis, the input and output bits of this S-box are represented as $(X₀, X₁, X₂, X₃)$, whereas in the specification, they are represented as $(X₃, X₂, X₁, X₀)$ where $Xᵢ ∈ \mathbb{F}_₂$. 

163
Table A.1: The S-box table of the AES and $s_1$ of ARIA

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63</td>
<td>7C</td>
<td>77</td>
<td>7B</td>
<td>F2</td>
<td>6B</td>
<td>6F</td>
<td>C5</td>
<td>30</td>
<td>01</td>
<td>67</td>
<td>2B</td>
<td>FE</td>
<td>D7</td>
<td>AB</td>
<td>76</td>
</tr>
<tr>
<td>1</td>
<td>CA</td>
<td>82</td>
<td>C9</td>
<td>7D</td>
<td>FA</td>
<td>59</td>
<td>47</td>
<td>F0</td>
<td>AD</td>
<td>D4</td>
<td>A2</td>
<td>AF</td>
<td>9C</td>
<td>A4</td>
<td>72</td>
<td>C0</td>
</tr>
<tr>
<td>2</td>
<td>B7</td>
<td>FD</td>
<td>93</td>
<td>26</td>
<td>36</td>
<td>3F</td>
<td>F7</td>
<td>CC</td>
<td>34</td>
<td>A5</td>
<td>E5</td>
<td>F1</td>
<td>71</td>
<td>D8</td>
<td>31</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>04</td>
<td>C7</td>
<td>23</td>
<td>C3</td>
<td>18</td>
<td>96</td>
<td>05</td>
<td>9A</td>
<td>07</td>
<td>12</td>
<td>80</td>
<td>E2</td>
<td>EB</td>
<td>27</td>
<td>B2</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>09</td>
<td>83</td>
<td>2C</td>
<td>1A</td>
<td>1B</td>
<td>6E</td>
<td>5A</td>
<td>A0</td>
<td>52</td>
<td>3B</td>
<td>D6</td>
<td>B3</td>
<td>29</td>
<td>E3</td>
<td>2F</td>
<td>84</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>D1</td>
<td>00</td>
<td>ED</td>
<td>2C</td>
<td>B1</td>
<td>5B</td>
<td>6A</td>
<td>CB</td>
<td>BE</td>
<td>39</td>
<td>44</td>
<td>4C</td>
<td>58</td>
<td>CF</td>
<td>62</td>
</tr>
<tr>
<td>6</td>
<td>D0</td>
<td>EF</td>
<td>AA</td>
<td>FB</td>
<td>43</td>
<td>4D</td>
<td>33</td>
<td>85</td>
<td>4F</td>
<td>92</td>
<td>02</td>
<td>7F</td>
<td>50</td>
<td>3C</td>
<td>9F</td>
<td>A8</td>
</tr>
<tr>
<td>7</td>
<td>51</td>
<td>A3</td>
<td>40</td>
<td>8F</td>
<td>92</td>
<td>9D</td>
<td>38</td>
<td>F5</td>
<td>BC</td>
<td>B6</td>
<td>DA</td>
<td>21</td>
<td>10</td>
<td>FF</td>
<td>F3</td>
<td>D2</td>
</tr>
<tr>
<td>8</td>
<td>CD</td>
<td>0C</td>
<td>13</td>
<td>EC</td>
<td>5F</td>
<td>97</td>
<td>44</td>
<td>17</td>
<td>C4</td>
<td>A7</td>
<td>7E</td>
<td>3D</td>
<td>64</td>
<td>5D</td>
<td>19</td>
<td>73</td>
</tr>
<tr>
<td>9</td>
<td>60</td>
<td>81</td>
<td>4F</td>
<td>DC</td>
<td>22</td>
<td>2A</td>
<td>90</td>
<td>88</td>
<td>46</td>
<td>EE</td>
<td>BB</td>
<td>14</td>
<td>DE</td>
<td>5E</td>
<td>08</td>
<td>DB</td>
</tr>
<tr>
<td>A</td>
<td>E0</td>
<td>32</td>
<td>3A</td>
<td>0A</td>
<td>49</td>
<td>06</td>
<td>24</td>
<td>5C</td>
<td>C2</td>
<td>D3</td>
<td>AC</td>
<td>62</td>
<td>91</td>
<td>95</td>
<td>E4</td>
<td>79</td>
</tr>
<tr>
<td>B</td>
<td>E7</td>
<td>C8</td>
<td>37</td>
<td>6D</td>
<td>8D</td>
<td>D5</td>
<td>4E</td>
<td>A9</td>
<td>6C</td>
<td>56</td>
<td>F4</td>
<td>EA</td>
<td>65</td>
<td>7A</td>
<td>AE</td>
<td>08</td>
</tr>
<tr>
<td>C</td>
<td>BA</td>
<td>78</td>
<td>25</td>
<td>2E</td>
<td>1C</td>
<td>A6</td>
<td>B4</td>
<td>C6</td>
<td>E8</td>
<td>DD</td>
<td>74</td>
<td>1F</td>
<td>4B</td>
<td>BD</td>
<td>8B</td>
<td>8A</td>
</tr>
<tr>
<td>D</td>
<td>70</td>
<td>3E</td>
<td>B5</td>
<td>66</td>
<td>48</td>
<td>03</td>
<td>F6</td>
<td>0E</td>
<td>61</td>
<td>35</td>
<td>57</td>
<td>B9</td>
<td>86</td>
<td>C1</td>
<td>1D</td>
<td>9E</td>
</tr>
<tr>
<td>E</td>
<td>E1</td>
<td>F8</td>
<td>98</td>
<td>11</td>
<td>99</td>
<td>8D</td>
<td>9E</td>
<td>94</td>
<td>9B</td>
<td>1E</td>
<td>87</td>
<td>E9</td>
<td>CE</td>
<td>55</td>
<td>26</td>
<td>DF</td>
</tr>
<tr>
<td>F</td>
<td>8C</td>
<td>A1</td>
<td>89</td>
<td>0D</td>
<td>BF</td>
<td>E6</td>
<td>42</td>
<td>68</td>
<td>41</td>
<td>99</td>
<td>2D</td>
<td>0F</td>
<td>B0</td>
<td>54</td>
<td>BB</td>
<td>16</td>
</tr>
</tbody>
</table>

Table A.2: The S-box $s_2$ table of ARIA

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>E2</td>
<td>4E</td>
<td>54</td>
<td>FC</td>
<td>94</td>
<td>C2</td>
<td>4A</td>
<td>CC</td>
<td>62</td>
<td>0D</td>
<td>6A</td>
<td>46</td>
<td>3C</td>
<td>4D</td>
<td>8B</td>
<td>D1</td>
</tr>
<tr>
<td>1</td>
<td>5E</td>
<td>FA</td>
<td>64</td>
<td>CB</td>
<td>B4</td>
<td>97</td>
<td>BE</td>
<td>2B</td>
<td>BC</td>
<td>77</td>
<td>2E</td>
<td>03</td>
<td>D3</td>
<td>19</td>
<td>59</td>
<td>C1</td>
</tr>
<tr>
<td>2</td>
<td>1D</td>
<td>06</td>
<td>41</td>
<td>6B</td>
<td>55</td>
<td>F0</td>
<td>99</td>
<td>69</td>
<td>EA</td>
<td>9C</td>
<td>18</td>
<td>AE</td>
<td>63</td>
<td>DF</td>
<td>E7</td>
<td>BB</td>
</tr>
<tr>
<td>3</td>
<td>00</td>
<td>73</td>
<td>66</td>
<td>FB</td>
<td>96</td>
<td>4C</td>
<td>85</td>
<td>E4</td>
<td>3A</td>
<td>09</td>
<td>45</td>
<td>AA</td>
<td>0F</td>
<td>EE</td>
<td>10</td>
<td>EB</td>
</tr>
<tr>
<td>4</td>
<td>2D</td>
<td>7F</td>
<td>F4</td>
<td>29</td>
<td>AC</td>
<td>CF</td>
<td>AD</td>
<td>91</td>
<td>8D</td>
<td>78</td>
<td>C8</td>
<td>95</td>
<td>F9</td>
<td>2F</td>
<td>CE</td>
<td>CD</td>
</tr>
<tr>
<td>5</td>
<td>08</td>
<td>7A</td>
<td>88</td>
<td>38</td>
<td>5C</td>
<td>83</td>
<td>2A</td>
<td>28</td>
<td>47</td>
<td>DB</td>
<td>B8</td>
<td>C7</td>
<td>93</td>
<td>A4</td>
<td>12</td>
<td>53</td>
</tr>
<tr>
<td>6</td>
<td>FF</td>
<td>87</td>
<td>0E</td>
<td>31</td>
<td>36</td>
<td>21</td>
<td>58</td>
<td>48</td>
<td>01</td>
<td>8E</td>
<td>37</td>
<td>74</td>
<td>32</td>
<td>CA</td>
<td>E9</td>
<td>B1</td>
</tr>
<tr>
<td>7</td>
<td>B7</td>
<td>AB</td>
<td>0C</td>
<td>D7</td>
<td>C4</td>
<td>56</td>
<td>42</td>
<td>26</td>
<td>07</td>
<td>98</td>
<td>60</td>
<td>D9</td>
<td>B6</td>
<td>90</td>
<td>11</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>EC</td>
<td>20</td>
<td>8C</td>
<td>BD</td>
<td>A0</td>
<td>C9</td>
<td>84</td>
<td>04</td>
<td>49</td>
<td>23</td>
<td>F1</td>
<td>4F</td>
<td>50</td>
<td>1F</td>
<td>13</td>
<td>DC</td>
</tr>
<tr>
<td>9</td>
<td>D8</td>
<td>C0</td>
<td>9E</td>
<td>57</td>
<td>E3</td>
<td>C3</td>
<td>7B</td>
<td>65</td>
<td>3B</td>
<td>02</td>
<td>8F</td>
<td>3E</td>
<td>E8</td>
<td>25</td>
<td>92</td>
<td>E5</td>
</tr>
<tr>
<td>A</td>
<td>15</td>
<td>DD</td>
<td>FD</td>
<td>17</td>
<td>A9</td>
<td>BF</td>
<td>D4</td>
<td>9A</td>
<td>7E</td>
<td>C5</td>
<td>39</td>
<td>67</td>
<td>FE</td>
<td>76</td>
<td>9D</td>
<td>43</td>
</tr>
<tr>
<td>B</td>
<td>A7</td>
<td>E1</td>
<td>D0</td>
<td>F5</td>
<td>68</td>
<td>F2</td>
<td>1B</td>
<td>34</td>
<td>70</td>
<td>05</td>
<td>A3</td>
<td>8A</td>
<td>D5</td>
<td>79</td>
<td>86</td>
<td>A8</td>
</tr>
<tr>
<td>C</td>
<td>30</td>
<td>C6</td>
<td>51</td>
<td>4B</td>
<td>1E</td>
<td>A6</td>
<td>27</td>
<td>F6</td>
<td>35</td>
<td>D2</td>
<td>6E</td>
<td>24</td>
<td>16</td>
<td>82</td>
<td>5F</td>
<td>DA</td>
</tr>
<tr>
<td>D</td>
<td>E6</td>
<td>75</td>
<td>A2</td>
<td>EF</td>
<td>2C</td>
<td>B2</td>
<td>1C</td>
<td>9F</td>
<td>5D</td>
<td>6F</td>
<td>80</td>
<td>0A</td>
<td>72</td>
<td>44</td>
<td>9B</td>
<td>6C</td>
</tr>
<tr>
<td>E</td>
<td>90</td>
<td>0B</td>
<td>58</td>
<td>33</td>
<td>7D</td>
<td>5A</td>
<td>52</td>
<td>F3</td>
<td>61</td>
<td>A1</td>
<td>F7</td>
<td>B0</td>
<td>D6</td>
<td>3F</td>
<td>7C</td>
<td>6D</td>
</tr>
<tr>
<td>F</td>
<td>ED</td>
<td>14</td>
<td>E0</td>
<td>A5</td>
<td>3D</td>
<td>22</td>
<td>B3</td>
<td>F8</td>
<td>89</td>
<td>DE</td>
<td>71</td>
<td>1A</td>
<td>AF</td>
<td>BA</td>
<td>B5</td>
<td>81</td>
</tr>
</tbody>
</table>

Table A.3: The S-box of Noekeon

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s[x]$</td>
<td>E</td>
<td>A</td>
<td>2</td>
<td>C</td>
<td>4</td>
<td>8</td>
<td>F</td>
<td>D</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>B</td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>
### Table A.4: The S-box of Present

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>s[x]</td>
<td>C</td>
<td>5</td>
<td>6</td>
<td>B</td>
<td>9</td>
<td>0</td>
<td>A</td>
<td>D</td>
<td>3</td>
<td>E</td>
<td>F</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table A.5: The eight S-boxes of Serpent

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀[x]</td>
<td>3</td>
<td>8</td>
<td>F</td>
<td>1</td>
<td>A</td>
<td>6</td>
<td>5</td>
<td>B</td>
<td>E</td>
<td>D</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td>C</td>
</tr>
<tr>
<td>s₁[x]</td>
<td>F</td>
<td>C</td>
<td>2</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>A</td>
<td>1</td>
<td>B</td>
<td>E</td>
<td>8</td>
<td>6</td>
<td>D</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>s₂[x]</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>C</td>
<td>A</td>
<td>F</td>
<td>D</td>
<td>1</td>
<td>E</td>
<td>4</td>
<td>0</td>
<td>B</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>s₃[x]</td>
<td>0</td>
<td>F</td>
<td>B</td>
<td>8</td>
<td>C</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>D</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>A</td>
<td>7</td>
<td>5</td>
<td>E</td>
</tr>
<tr>
<td>s₄[x]</td>
<td>1</td>
<td>F</td>
<td>8</td>
<td>3</td>
<td>C</td>
<td>0</td>
<td>B</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>A</td>
<td>9</td>
<td>E</td>
<td>7</td>
<td>D</td>
</tr>
<tr>
<td>s₅[x]</td>
<td>F</td>
<td>5</td>
<td>2</td>
<td>B</td>
<td>A</td>
<td>9</td>
<td>C</td>
<td>0</td>
<td>3</td>
<td>E</td>
<td>D</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>s₆[x]</td>
<td>7</td>
<td>2</td>
<td>C</td>
<td>5</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>B</td>
<td>E</td>
<td>9</td>
<td>1</td>
<td>F</td>
<td>D</td>
<td>3</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>s₇[x]</td>
<td>1</td>
<td>D</td>
<td>F</td>
<td>0</td>
<td>E</td>
<td>8</td>
<td>2</td>
<td>B</td>
<td>7</td>
<td>4</td>
<td>C</td>
<td>A</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

### Table A.6: The S-box table of SMS4

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | D6 | 90 | E9 | FE | CC | E1 | 3D | B7 | 16 | B6 | 14 | C2 | 28 | FB | 2C | 05 |
| 1 | 2B | 67 | 9A | 76 | 2A | BE | 04 | C3 | AA | 44 | 13 | 26 | 49 | 86 | 06 | 99 |
| 2 | 9C | 42 | 50 | F4 | 91 | EF | 98 | 7A | 33 | 54 | 0B | 43 | ED | CF | AC | 62 |
| 3 | E4 | B3 | 1C | A9 | C9 | 08 | E8 | 95 | 80 | DF | 94 | FA | 75 | 8F | 3F | A6 |
| 4 | 47 | 07 | A7 | FC | F3 | 73 | 17 | BA | 83 | 59 | 3C | 19 | E6 | 85 | 4F | A8 |
| 5 | 68 | 6B | 81 | B2 | 71 | 64 | DA | 8B | F8 | EB | 0F | 4B | 70 | 56 | 9D | 35 |
| 6 | 1E | 24 | 0E | 5E | 63 | 58 | D1 | A2 | 25 | 22 | 7C | 3B | 01 | 21 | 78 | 87 |
| 7 | D4 | 00 | 46 | 57 | 9F | D3 | 27 | 52 | 4C | 36 | 02 | E7 | A0 | C4 | C8 | 9E |
| 8 | EA | BF | 8A | D2 | 40 | C7 | 38 | B5 | A3 | F7 | F2 | CE | F9 | 61 | 15 | A1 |
| 9 | E0 | AE | 5D | A4 | 9B | 34 | 1A | 55 | AD | 93 | 32 | 30 | F5 | 8C | B1 | E3 |
| A | 1D | F6 | E2 | 2E | 82 | 66 | CA | 60 | C0 | 29 | 23 | AB | 0D | 53 | 4D | 6F |
| B | D5 | DB | 37 | 45 | DE | FD | 8E | 2F | 03 | FF | 6A | 72 | 6D | 6C | 5B | 51 |
| C | 8D | 1B | AF | 92 | BB | DD | BC | 7F | 11 | D9 | 5C | 41 | 1F | 10 | 5A | D8 |
| D | 0A | C1 | 31 | 88 | A5 | CD | 7B | BD | 2D | 74 | D0 | 12 | B8 | E5 | B4 | B0 |
| E | 89 | 69 | 97 | 4A | 0C | 96 | 77 | 7E | 65 | B9 | F1 | 09 | C5 | 6E | C6 | 84 |
| F | 18 | F0 | 7D | EC | 3A | DC | 4D | 20 | 79 | EE | 5F | 3E | D7 | CB | 39 | 48 |
Appendix B

Difference Distribution Tables

This appendix contains the difference distribution tables (DDT) for the $4 \times 4$ S-boxes of the block ciphers Noekeon, Serpent and PRESENT. These tables are referred to in Chapter 3. The technique to generate these tables is given in Section 2.3.2.
### Table B.1: The Difference Distribution Table of the S-box of Noekeon

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table B.2: The Difference Distribution Table of the S-box $s_0$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Table B.3: The Difference Distribution Table of the S-box $s_1$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

Table B.4: The Difference Distribution Table of the S-box $s_2$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
### Table B.5: The Difference Distribution Table of the S-box $s_3$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

### Table B.6: The Difference Distribution Table of the S-box $s_4$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Table B.7: The Difference Distribution Table of the S-box $s_5$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

Table B.8: The Difference Distribution Table of the S-box $s_6$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>A</td>
<td>.</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>B</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>D</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>2</td>
<td>4</td>
<td>.</td>
<td>2</td>
<td>.</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.</td>
</tr>
<tr>
<td>F</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

### Table B.9: The Difference Distribution Table of the S-box $s_7$ of Serpent

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table B.10: The Difference Distribution Table of the S-box of PRESENT

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Appendix C

LEX-AES Equations

C.1 Equations

The following list the relations of the temporary variables $p_i,q_i,r_i,s_i,t_i,u_i$ in the forward and backward directions. The terms $c_i$, $x_i$ and $k_i$ denote the keystream (which are the constants), internal state and subkey bytes, respectively. The temporary variables are substituted so that only the $x_i$ and $k_i$ variables remain. In the following equations, the functions $\Theta$ and $\Pi$ are defined as follows:

$$\Theta(z_0, z_1, z_2, z_3) = 2s[z_0] \oplus 3s[z_1] \oplus s[z_2] \oplus s[z_3],$$
$$\Pi(z_0, z_1, z_2, z_3) = Ez_0 \oplus Bz_1 \oplus Dz_2 \oplus 9z_3.$$

C.1.1 Forward Direction

- $s_0 = \Theta(c_{16}, x_3, c_{19}, x_{11}) \oplus k_{64}$
- $s_1 = \Theta(x_3, c_{19}, x_{11}, c_{16}) \oplus k_{65}$
- $s_2 = \Theta(c_{19}, x_{11}, c_{16}, x_3) \oplus k_{66}$
- $s_3 = \Theta(x_{11}, c_{16}, x_3, c_{19}) \oplus k_{67}$
- $s_4 = \Theta(x_6, x_{10}, x_1, x_2) \oplus k_{69}$
- $s_5 = \Theta(x_1, x_2, x_6, x_{10}) \oplus k_{71}$
• \( s_6 = \Theta(c_{18}, x_9, c_{17}, x_5) \oplus k_{72} \)
• \( s_7 = \Theta(x_9, c_{17}, x_5, c_{18}) \oplus k_{73} \)
• \( s_8 = \Theta(c_{17}, x_5, c_{18}, x_9) \oplus k_{74} \)
• \( s_9 = \Theta(x_5, c_{18}, x_9, c_{17}) \oplus k_{75} \)
• \( s_{10} = \Theta(x_0, x_4, x_7, x_8) \oplus k_{77} \)
• \( s_{11} = \Theta(x_7, x_8, x_0, x_4) \oplus k_{79} \)
• \( t_0 = \Theta(s_4, s_8, s_{11}, s_0) \oplus k_{81} \)
• \( t_1 = \Theta(s_{11}, s_0, s_4, s_8) \oplus k_{83} \)
• \( t_2 = \Theta(c_{20}, s_7, c_{23}, s_3) \oplus k_{84} \)
• \( t_3 = \Theta(s_7, c_{23}, s_3, c_{20}) \oplus k_{85} \)
• \( t_4 = \Theta(c_{23}, s_3, c_{20}, s_7) \oplus k_{86} \)
• \( t_5 = \Theta(s_3, c_{20}, s_7, c_{23}) \oplus k_{87} \)
• \( t_6 = \Theta(s_{10}, s_2, s_5, s_6) \oplus k_{89} \)
• \( t_7 = \Theta(s_5, s_6, s_{10}, s_2) \oplus k_{91} \)
• \( t_8 = \Theta(c_{22}, s_1, c_{21}, s_9) \oplus k_{92} \)
• \( t_9 = \Theta(s_1, c_{21}, s_9, c_{22}) \oplus k_{93} \)
• \( t_{10} = \Theta(c_{21}, s_9, c_{22}, s_1) \oplus k_{94} \)
• \( t_{11} = \Theta(s_9, c_{22}, s_1, c_{21}) \oplus k_{95} \)
• \( u_0 = \Theta(c_{24}, t_3, c_{27}, t_{11}) \oplus k_{96} \)
• \( u_1 = \Theta(t_3, c_{27}, t_{11}, c_{24}) \oplus k_{97} \)
• \( u_2 = \Theta(c_{27}, t_{11}, c_{24}, t_3) \oplus k_{98} \)
• \( u_3 = \Theta(t_{11}, c_{24}, t_3, c_{27}) \oplus k_{99} \)
• \( u_4 = \Theta(t_6, t_{10}, t_1, t_2) \oplus k_{101} \)
C.1. Equations

\begin{itemize}
  \item \( u_5 = \Theta(t_1, t_2, t_6, t_{10}) \oplus k_{103} \)
  \item \( u_6 = \Theta(c_{26}, t_9, c_{25}, t_5) \oplus k_{104} \)
  \item \( u_7 = \Theta(t_9, c_{25}, t_5, c_{26}) \oplus k_{105} \)
  \item \( u_8 = \Theta(c_{25}, t_5, c_{26}, t_9) \oplus k_{106} \)
  \item \( u_9 = \Theta(t_5, c_{26}, t_9, c_{25}) \oplus k_{107} \)
  \item \( u_{10} = \Theta(t_0, t_4, t_7, t_8) \oplus k_{109} \)
  \item \( u_{11} = \Theta(t_7, t_8, t_0, t_4) \oplus k_{111} \)
\end{itemize}

C.1.2 Backward Direction

\begin{itemize}
  \item \( r_0 = S^{-1}[\Pi(c_{16}, x_0, c_{17}, x_1) \oplus \Pi(k_{48}, k_{49}, k_{50}, k_{51})] \)
  \item \( r_1 = S^{-1}[\Pi(x_9, x_{10}, x_{11}, x_8) \oplus \Pi(k_{61}, k_{62}, k_{63}, k_{60})] \)
  \item \( r_2 = S^{-1}[\Pi(c_{19}, x_7, c_{18}, x_6) \oplus \Pi(k_{58}, k_{59}, k_{56}, k_{57})] \)
  \item \( r_3 = S^{-1}[\Pi(x_5, x_2, x_3, x_4) \oplus \Pi(k_{55}, k_{52}, k_{53}, k_{54})] \)
  \item \( r_4 = S^{-1}[\Pi(x_0, c_{17}, x_1, c_{16}) \oplus \Pi(k_{49}, k_{50}, k_{51}, k_{48})] \)
  \item \( r_5 = S^{-1}[\Pi(x_7, c_{18}, x_6, c_{19}) \oplus \Pi(k_{59}, k_{56}, k_{57}, k_{58})] \)
  \item \( r_6 = S^{-1}[\Pi(c_{18}, x_6, c_{19}, x_7) \oplus \Pi(k_{56}, k_{57}, k_{58}, k_{59})] \)
  \item \( r_7 = S^{-1}[\Pi(x_3, x_4, x_5, x_2) \oplus \Pi(k_{53}, k_{54}, k_{55}, k_{52})] \)
  \item \( r_8 = S^{-1}[\Pi(c_{17}, x_1, c_{16}, x_0) \oplus \Pi(k_{50}, k_{51}, k_{48}, k_{49})] \)
  \item \( r_9 = S^{-1}[\Pi(x_{11}, x_8, x_9, x_{10}) \oplus \Pi(k_{63}, k_{60}, k_{61}, k_{62})] \)
  \item \( r_{10} = S^{-1}[\Pi(x_6, c_{19}, x_7, c_{18}) \oplus \Pi(k_{57}, k_{58}, k_{59}, k_{56})] \)
  \item \( r_{11} = S^{-1}[\Pi(x_1, c_{16}, x_0, c_{17}) \oplus \Pi(k_{51}, k_{48}, k_{49}, k_{50})] \)
  \item \( q_0 = S^{-1}[\Pi(r_{10}, c_{15}, r_{11}, c_{14}) \oplus \Pi(k_{45}, k_{46}, k_{47}, k_{44})] \)
  \item \( q_1 = S^{-1}[\Pi(r_5, c_{12}, r_4, c_{13}) \oplus \Pi(k_{39}, k_{36}, k_{37}, k_{38})] \)
  \item \( q_2 = S^{-1}[\Pi(c_{12}, r_4, c_{13}, r_5) \oplus \Pi(k_{36}, k_{37}, k_{38}, k_{39})] \)
\end{itemize}
• \( q_3 = S^{-1}[\Pi(r_1, r_2, r_3, r_0) \oplus \Pi(k_{33}, k_{34}, k_{35}, k_{32})] \)
• \( q_4 = S^{-1}[\Pi(c_{15}, r_{11}, c_{14}, r_{10}) \oplus \Pi(k_{46}, k_{47}, k_{44}, k_{45})] \)
• \( q_5 = S^{-1}[\Pi(r_9, r_6, r_7, r_8) \oplus \Pi(k_{43}, k_{46}, k_{44}, k_{42})] \)
• \( q_6 = S^{-1}[\Pi(r_4, c_{13}, r_5, c_{12}) \oplus \Pi(k_{37}, k_{38}, k_{39}, k_{36})] \)
• \( q_7 = S^{-1}[\Pi(r_{11}, c_{14}, r_{10}, c_{15}) \oplus \Pi(k_{47}, k_{44}, k_{45}, k_{46})] \)
• \( q_8 = S^{-1}[\Pi(c_{14}, r_{10}, c_{15}, r_{11}) \oplus \Pi(k_{44}, k_{45}, k_{16}, k_{47})] \)
• \( q_9 = S^{-1}[\Pi(r_7, r_8, r_9, r_6) \oplus \Pi(k_{41}, k_{42}, k_{43}, k_{40})] \)
• \( q_{10} = S^{-1}[\Pi(c_{13}, r_5, c_{12}, r_4) \oplus \Pi(k_{38}, k_{39}, k_{36}, k_{37})] \)
• \( q_{11} = S^{-1}[\Pi(r_3, r_0, r_1, r_2) \oplus \Pi(k_{35}, k_{32}, k_{33}, k_{34})] \)
• \( p_0 = S^{-1}[\Pi(c_8, q_0, c_9, q_1) \oplus \Pi(k_{16}, k_{17}, k_{18}, k_{19})] \)
• \( p_1 = S^{-1}[\Pi(q_9, q_{10}, q_{11}, q_8) \oplus \Pi(k_{29}, k_{30}, k_{31}, k_{28})] \)
• \( p_2 = S^{-1}[\Pi(c_{11}, q_7, c_{10}, q_6) \oplus \Pi(k_{26}, k_{27}, k_{24}, k_{25})] \)
• \( p_3 = S^{-1}[\Pi(q_5, q_2, q_3, q_4) \oplus \Pi(k_{23}, k_{20}, k_{21}, k_{22})] \)
• \( p_4 = S^{-1}[\Pi(q_0, c_9, q_1, c_8) \oplus \Pi(k_{17}, k_{18}, k_{19}, k_{16})] \)
• \( p_5 = S^{-1}[\Pi(q_{10}, c_6, q_{11}, c_1) \oplus \Pi(k_{27}, k_{24}, k_{25}, k_{26})] \)
• \( p_6 = S^{-1}[\Pi(c_{10}, q_6, c_{11}, q_7) \oplus \Pi(k_{24}, k_{25}, k_{26}, k_{27})] \)
• \( p_7 = S^{-1}[\Pi(q_3, q_4, q_5, q_2) \oplus \Pi(k_{21}, k_{22}, k_{23}, k_{20})] \)
• \( p_8 = S^{-1}[\Pi(c_9, q_1, c_8, q_0) \oplus \Pi(k_{18}, k_{19}, k_{16}, k_{17})] \)
• \( p_9 = S^{-1}[\Pi(q_{11}, q_8, q_9, q_{10}) \oplus \Pi(k_{31}, k_{28}, k_{29}, k_{30})] \)
• \( p_{10} = S^{-1}[\Pi(q_6, c_{11}, q_7, c_{10}) \oplus \Pi(k_{25}, k_{26}, k_{27}, k_{24})] \)
• \( p_{11} = S^{-1}[\Pi(q_1, c_8, q_0, c_9) \oplus \Pi(k_{19}, k_{16}, k_{17}, k_{18})] \)
C.1.3 Subkey Variables Substitution

The following substitutions are performed so that only 16 subkey variables remain.

- $k_0 = k_{48} \oplus S[k_{61} \oplus k_{57}] \oplus S[k_{61} \oplus k_{53}] \oplus S[k_{61} \oplus k_{57} \oplus k_{49}] \oplus RC^1 \oplus RC^2 \oplus RC^3$
- $k_1 = k_{49} \oplus S[k_{62} \oplus k_{58}] \oplus S[k_{62} \oplus k_{54}] \oplus S[k_{62} \oplus k_{54} \oplus k_{58} \oplus k_{50}]$
- $k_2 = k_{50} \oplus S[k_{63} \oplus k_{59}] \oplus S[k_{63} \oplus k_{55}] \oplus S[k_{63} \oplus k_{55} \oplus k_{59} \oplus k_{51}]$
- $k_3 = k_{51} \oplus S[k_{60} \oplus k_{56}] \oplus S[k_{60} \oplus k_{52}] \oplus S[k_{60} \oplus k_{52} \oplus k_{56} \oplus k_{48}]$
- $k_4 = k_{48} \oplus k_{52} \oplus S[k_{61} \oplus k_{53}] \oplus RC^2$
- $k_5 = k_{49} \oplus k_{53} \oplus S[k_{62} \oplus k_{54}]$
- $k_6 = k_{50} \oplus k_{54} \oplus S[k_{63} \oplus k_{55}]$
- $k_7 = k_{51} \oplus k_{55} \oplus S[k_{60} \oplus k_{52}]$
- $k_8 = k_{48} \oplus k_{52} \oplus k_{56} \oplus S[k_{61} \oplus k_{57}] \oplus RC^3$
- $k_9 = k_{49} \oplus k_{53} \oplus k_{57} \oplus S[k_{62} \oplus k_{58}]$
- $k_{10} = k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63} \oplus k_{59}]$
- $k_{11} = k_{51} \oplus k_{55} \oplus k_{59} \oplus S[k_{60} \oplus k_{56}]$
- $k_{12} = k_{48} \oplus k_{52} \oplus k_{56} \oplus k_{60}$
- $k_{13} = k_{49} \oplus k_{53} \oplus k_{57} \oplus k_{61}$
- $k_{14} = k_{50} \oplus k_{54} \oplus k_{58} \oplus k_{62}$
- $k_{15} = k_{51} \oplus k_{55} \oplus k_{59} \oplus k_{63}$
- $k_{16} = k_{48} \oplus S[k_{61} \oplus k_{57}] \oplus S[k_{61} \oplus k_{53}] \oplus RC^2 \oplus RC^3$
- $k_{17} = k_{49} \oplus S[k_{62} \oplus k_{58}] \oplus S[k_{62} \oplus k_{54}]$
- $k_{18} = k_{50} \oplus S[k_{63} \oplus k_{59}] \oplus S[k_{63} \oplus k_{55}]$
- $k_{19} = k_{51} \oplus S[k_{60} \oplus k_{56}] \oplus S[k_{60} \oplus k_{52}]$
- $k_{20} = k_{52} \oplus S[k_{61} \oplus k_{57}] \oplus RC^3$
Appendix C. LEX-AES Equations

- \(k_{21} = k_{53} \oplus S[k_{62} \oplus k_{58}]\)
- \(k_{22} = k_{54} \oplus S[k_{63} \oplus k_{59}]\)
- \(k_{23} = k_{55} \oplus S[k_{60} \oplus k_{56}]\)
- \(k_{24} = k_{48} \oplus k_{50}\)
- \(k_{25} = k_{49} \oplus k_{57}\)
- \(k_{26} = k_{50} \oplus k_{58}\)
- \(k_{27} = k_{51} \oplus k_{59}\)
- \(k_{28} = k_{52} \oplus k_{60}\)
- \(k_{29} = k_{53} \oplus k_{61}\)
- \(k_{30} = k_{54} \oplus k_{62}\)
- \(k_{31} = k_{55} \oplus k_{63}\)
- \(k_{32} = k_{48} \oplus S[k_{61} \oplus k_{57}] \oplus RC^3\)
- \(k_{33} = k_{49} \oplus S[k_{62} \oplus k_{58}]\)
- \(k_{34} = k_{50} \oplus S[k_{63} \oplus k_{59}]\)
- \(k_{35} = k_{51} \oplus S[k_{60} \oplus k_{56}]\)
- \(k_{36} = k_{48} \oplus k_{52}\)
- \(k_{37} = k_{49} \oplus k_{53}\)
- \(k_{38} = k_{50} \oplus k_{54}\)
- \(k_{39} = k_{51} \oplus k_{55}\)
- \(k_{40} = k_{52} \oplus k_{56}\)
- \(k_{41} = k_{53} \oplus k_{57}\)
- \(k_{42} = k_{54} \oplus k_{58}\)
- \(k_{43} = k_{55} \oplus k_{59}\)
• $k_{44} = k_{56} \oplus k_{60}$
• $k_{45} = k_{57} \oplus k_{61}$
• $k_{46} = k_{58} \oplus k_{62}$
• $k_{47} = k_{59} \oplus k_{63}$
• $k_{64} = k_{48} \oplus S[k_{61}] \oplus RC^4$
• $k_{65} = k_{49} \oplus S[k_{62}]$
• $k_{66} = k_{50} \oplus S[k_{63}]$
• $k_{67} = k_{51} \oplus S[k_{60}]$
• $k_{68} = k_{48} \oplus k_{52} \oplus S[k_{61}] \oplus RC^4$
• $k_{69} = k_{49} \oplus k_{53} \oplus S[k_{62}]$
• $k_{70} = k_{50} \oplus k_{54} \oplus S[k_{63}]$
• $k_{71} = k_{51} \oplus k_{55} \oplus S[k_{60}]$
• $k_{72} = k_{48} \oplus k_{52} \oplus k_{56} \oplus S[k_{61}] \oplus RC^4$
• $k_{73} = k_{49} \oplus k_{53} \oplus k_{57} \oplus S[k_{62}]$
• $k_{74} = k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}]$
• $k_{75} = k_{51} \oplus k_{55} \oplus k_{59} \oplus S[k_{60}]$
• $k_{76} = k_{48} \oplus k_{52} \oplus k_{56} \oplus k_{60} \oplus S[k_{61}] \oplus RC^4$
• $k_{77} = k_{49} \oplus k_{53} \oplus k_{57} \oplus k_{61} \oplus S[k_{62}]$
• $k_{78} = k_{50} \oplus k_{54} \oplus k_{58} \oplus k_{62} \oplus S[k_{63}]$
• $k_{79} = k_{51} \oplus k_{55} \oplus k_{59} \oplus k_{63} \oplus S[k_{60}]$
• $k_{80} = k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^4 \oplus RC^5$
• $k_{81} = k_{49} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]]$
• $k_{82} = k_{50} \oplus S[k_{63}] \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]]$
\[ k_{33} = k_{51} \oplus S[k_{60}] \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \]

\[ k_{34} = k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^5 \]

\[ k_{35} = k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]] \]

\[ k_{36} = k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]] \]

\[ k_{37} = k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \]

\[ k_{38} = k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^4 \oplus RC^5 \]

\[ k_{39} = k_{49} \oplus k_{57} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]] \]

\[ k_{40} = k_{50} \oplus k_{58} \oplus S[k_{63}] \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]] \]

\[ k_{41} = k_{51} \oplus k_{59} \oplus S[k_{60}] \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \]

\[ k_{42} = k_{52} \oplus k_{60} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^5 \]

\[ k_{43} = k_{53} \oplus k_{61} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]] \]

\[ k_{44} = k_{54} \oplus k_{62} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]] \]

\[ k_{45} = k_{55} \oplus k_{63} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \]

\[ k_{46} = k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]]] \oplus RC^4 \oplus RC^5 \oplus RC^6 \]

\[ k_{47} = k_{49} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]] \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]]] \]

\[ k_{48} = k_{50} \oplus S[k_{63}] \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]] \oplus S[k_{63} + k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4]] \]

\[ k_{49} = k_{51} \oplus S[k_{60}] \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^5] \]

\[ k_{50} = k_{48} \oplus k_{52} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus S[k_{63}]]] \oplus RC^4 \oplus RC^6 \]

\[ k_{51} = k_{49} \oplus k_{53} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]]] \]

\[ k_{52} = k_{50} \oplus k_{54} \oplus S[k_{63}] \oplus S[k_{63} \oplus k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4]] \]
• $k_{103} = k_{51} \oplus k_{55} \oplus S[k_{60}] \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5]$

• $k_{104} = k_{52} \oplus k_{56} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]]) \oplus RC^5 \oplus RC^6$

• $k_{105} = k_{53} \oplus k_{57} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{56} \oplus S[k_{63}] \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]])$

• $k_{106} = k_{54} \oplus k_{58} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}] \oplus S[k_{63} \oplus k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4]]$

• $k_{107} = k_{55} \oplus k_{59} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5]$

• $k_{108} = k_{56} \oplus k_{60} \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]] \oplus RC^6$

• $k_{109} = k_{57} \oplus k_{61} \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]]$

• $k_{110} = k_{58} \oplus k_{62} \oplus S[k_{63} \oplus k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4]]$

• $k_{111} = k_{59} \oplus k_{63} \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5]$

• $k_{112} = k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]]) \oplus S[k_{61} \oplus k_{57} \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]] \oplus RC^4 \oplus RC^5 \oplus RC^6 \oplus RC^7$

• $k_{113} = k_{49} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}] \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]] \oplus S[k_{62} \oplus k_{58} \oplus S[k_{63} \oplus k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4]]$

• $k_{114} = k_{50} \oplus S[k_{63}] \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}] \oplus S[k_{63} \oplus k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4]] \oplus S[k_{63} \oplus k_{59} \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5]]$

• $k_{115} = k_{51} \oplus S[k_{60}] \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5] \oplus S[k_{60} \oplus k_{56} \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}]]) \oplus RC^6]$

• $k_{116} = k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus S[k_{61} \oplus k_{57} \oplus S[k_{62} \oplus k_{54} \oplus S[k_{63} \oplus k_{59} \oplus k_{55} \oplus k_{51} \oplus S[k_{60}]]) \oplus RC^5 \oplus RC^7$
C.1.4 Temporary Variables

Temporary variables from Section 4.2.3 are defined here. Note that multiplication is performed in the AES $GF(2^8)$ field and the coefficients are in hexadecimal format.

- $v_0 = \Theta(c_{16}, 47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus CS[c_{15}], c_{19}, CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus CS[c_{14}] \oplus 44S[c_{13}]) \oplus k_{48} \oplus S[k_{61}] \oplus RC^4$
- $v_1 = 44S[x_2] \oplus CS[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBC_{21}$
C.1. Equations

\[ v_2 = \Theta(c_{17}, C B x_{2} \oplus 47 x_4 \oplus C B k_{52} \oplus 47 k_{54} \oplus k_{55} \oplus C 9 S[c_{12}] \oplus 44 S[c_{15}], c_{18}, 47 x_8 \oplus C B x_{10} \oplus 47 k_{60} \oplus k_{61} \oplus C B k_{62} \oplus 44 S[c_{14}] \oplus C 9 S[c_{13}]) \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \]

\[ v_3 = C 9 S[x_8] \oplus 44 S[x_4] \oplus C B k_{18} \oplus 47 k_{50} \oplus k_{51} \oplus C B k_{52} \oplus 47 k_{54} \oplus k_{55} \oplus C B k_{56} \oplus 47 k_{58} \oplus C B k_{60} \oplus S[k_{60}] \oplus C B S[k_{61}] \oplus 47 k_{62} \oplus k_{63} \oplus 47 S[k_{63}] \oplus C B R C^4 \oplus C B c_{22} \oplus 47 c_{23} \]

\[ w_0 = \Theta(c_{18}, 47 x_8 \oplus C B x_{10} \oplus 47 k_{60} \oplus k_{61} \oplus C B k_{62} \oplus 44 S[c_{14}] \oplus C 9 S[c_{13}], S[c_{17}], C B x_2 \oplus 47 x_4 \oplus C B k_{52} \oplus 47 k_{54} \oplus k_{55} \oplus C 9 S[c_{12}] \oplus 44 S[c_{15}] \oplus k_{48} \oplus k_{52} \oplus k_{56} \oplus S[k_{61}] \oplus R C^4 \]

\[ w_1 = 44 S[x_8] \oplus C 9 S[x_4] \oplus 47 k_{48} \oplus k_{49} \oplus C B k_{50} \oplus 47 k_{52} \oplus k_{53} \oplus C B k_{54} \oplus 47 k_{56} \oplus k_{57} \oplus C B k_{58} \oplus 47 k_{60} \oplus k_{61} \oplus 47 S[k_{61}] \oplus C B k_{62} \oplus S[k_{62}] \oplus C B S[k_{63}] \oplus 47 R C^4 \oplus 47 c_{22} \oplus C B c_{23} \]

\[ w_2 = \Theta(c_{19}, C B x_8 \oplus 47 x_{10} \oplus C B k_{60} \oplus 47 k_{62} \oplus k_{63} \oplus C 9 S[c_{14}] \oplus 44 S[c_{13}], c_{16}, 47 x_2 \oplus C B x_4 \oplus 47 k_{52} \oplus k_{53} \oplus C B k_{54} \oplus 44 S[c_{12}] \oplus C 9 S[c_{15}] \oplus k_{50} \oplus S[k_{63}] \]

\[ w_3 = C 9 S[x_2] \oplus 44 S[x_{10}] \oplus C B k_{48} \oplus 47 k_{50} \oplus C B k_{52} \oplus 47 k_{54} \oplus k_{55} \oplus S[k_{60}] \oplus C B S[k_{61}] \oplus 47 S[k_{63}] \oplus C B R C^4 \oplus C B c_{20} \oplus 47 c_{21} \]

\[ y_0 = 7 S[2 S[c_{16}] \oplus 3 S[47 x_2 \oplus C B x_4 \oplus 47 k_{52} \oplus k_{53} \oplus C B k_{54} \oplus 44 S[c_{12}] \oplus C 9 S[c_{15}] \oplus S[c_{19}] \oplus S[C B x_8 \oplus 47 x_{10} \oplus C B k_{60} \oplus 47 k_{62} \oplus k_{63} \oplus C 9 S[c_{14}] \oplus 44 S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus R C^4] \]

\[ y_1 = 4 S[44 S[x_2] \oplus C 9 S[x_{10}] \oplus 47 k_{48} \oplus k_{49} \oplus C B k_{50} \oplus 47 k_{52} \oplus k_{53} \oplus C B k_{54} \oplus 47 S[k_{61}] \oplus S[k_{62}] \oplus C B S[k_{63}] \oplus 47 R C^4 \oplus 47 c_{20} \oplus C B c_{21}] \]

\[ y_2 = S[S[c_{18}] \oplus S[47 x_8 \oplus C B x_{10} \oplus 47 k_{60} \oplus k_{61} \oplus C B k_{62} \oplus 44 S[c_{14}] \oplus C 9 S[c_{13}]] \oplus 2 S[c_{17}] \oplus 3 S[C B x_2 \oplus 47 x_4 \oplus C B k_{52} \oplus 47 k_{54} \oplus k_{55} \oplus C 9 S[c_{12}] \oplus 44 S[c_{15}]] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \]

\[ y_3 = 44 S[x_4] \oplus C 9 S[x_8] \oplus C B k_{48} \oplus 47 k_{50} \oplus C B k_{52} \oplus 47 k_{54} \oplus 47 k_{58} \oplus C B k_{60} \oplus C B S[k_{61}] \oplus k_{63} \oplus 47 S[k_{63}] \oplus C B R C^4 \oplus 47 k_{62} \oplus k_{63} \oplus C B c_{22} \oplus 47 c_{23} \oplus S^{-1}[2 S[2 S[c_{16}] \oplus 3 S[47 x_2 \oplus C B x_4 \oplus 47 k_{52} \oplus k_{53} \oplus C B k_{54} \oplus 44 S[c_{12}] \oplus C 9 S[c_{15}] \oplus S[c_{19}] \oplus S[C B x_8 \oplus 47 x_{10} \oplus C B k_{60} \oplus 47 k_{62} \oplus k_{63} \oplus C 9 S[c_{14}] \oplus 44 S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus R C^4] \oplus 3 S[44 S[x_2] \oplus C 9 S[x_{10}] \oplus 47 k_{48} \oplus k_{49} \oplus C B k_{50} \oplus 47 k_{52} \oplus k_{53} \oplus C B k_{54} \oplus 47 S[k_{61}] \oplus S[k_{62}] \oplus C B S[k_{63}] \oplus 47 R C^4 \oplus 47 c_{20} \oplus C B c_{21}] \oplus S[S[c_{18}] \oplus S[47 x_8 \oplus C B x_{10} \oplus 47 k_{60} \oplus k_{61} \oplus C B k_{62} \oplus 44 S[c_{14}] \oplus C 9 S[c_{13}]] \oplus 2 S[c_{17}] \oplus 3 S[C B x_2 \oplus 47 x_4 \oplus C B k_{52} \oplus 47 k_{54} \oplus k_{55} \oplus C 9 S[c_{12}] \oplus 44 S[c_{15}]] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus R C^4 \oplus R C^5 \oplus c_{24}] \]
The temporary key variables are:

- \( K_0^* = k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^4 \oplus RC^5 \)
- \( K_1^* = k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^4 \oplus RC^5 \)
- \( K_2^* = 3k_{48} \oplus k_{50} \oplus 3S[k_{61}] \oplus S[k_{63}] \oplus 3S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus 3RC^4 \oplus 3RC^5 \oplus 3c_{24} \)

### C.1.5 Final Equation System

The final system of 21 equations in 17 variables are given here. The expressions \( W_j, Y_j, Z_j \) and \( T_j \) contain subkey and/or key variables which are defined later. Note that \( \Theta \) is defined in Section 4.2.1, and \( \hat{\Pi}(z_0, z_1, z_2, z_3) = ES^{-1}[z_0] \oplus BS^{-1}[z_1] \oplus BS^{-1}[z_2] \oplus 9S^{-1}[z_3] \). Note that multiplication is performed in the AES \( GF(2^8) \) field and the coefficients are in hexadecimal format.

1. Based on Equation (4.25):

\[
\begin{align*}
\Theta(\Theta( & c_{24}, \Theta(Y_0, c_{23}, Y_1, c_{20}) \oplus T_0, \\
c_{27}, \Theta(Y_3, c_{22}, Y_2, c_{21}) \oplus T_1) \oplus T_2, \\
\Theta( & W_2, W_0, W_1, 0) \oplus T_4, \\
\Theta(c_{21}, Y_3, c_{22}, Y_2) \oplus T_5, \\
7S[W_4] & \oplus 7S[W_5] \oplus 3S[W_6] \oplus T_6 \oplus T_{61}, \\
\Theta(c_{20}, Y_0, c_{23}, Y_1) & \oplus T_3) \oplus T_7, \\
\Theta( & c_{25}, \Theta(Y_1, c_{20}, Y_0, c_{23}) \oplus T_9, \\
c_{26}, \Theta(Y_2, c_{21}, Y_3, c_{22}) & \oplus T_8) \oplus T_{10} \\
7S[W_0] & \oplus 7S[W_1] \oplus 3S[W_2] \oplus T_{14} \oplus T_{62}, \\
\Theta(c_{22}, Y_2, c_{21}, Y_3) & \oplus T_{11}, \\
3S[W_4] & \oplus S[W_5] \oplus 2S[W_6] \oplus T_{12}, \\
\Theta(c_{23}, Y_1, c_{20}, Y_0) & \oplus T_{13}) \oplus T_{15}) \\
\Theta( & T_{16} \oplus c_{32} = 0 
\end{align*}
\]
2. Based on Equation (4.26):

\[
\Theta(\Theta(c_{25}, \Theta(Y_1, c_{20}, Y_0, c_{23}) \oplus T_9, c_{26}, \Theta(Y_2, c_{21}, Y_3, c_{22}) \oplus T_8) \oplus T_{10},
\]

\[
\Theta(7S[W_0] \oplus 7S[W_1] \oplus 3S[W_2] \oplus T_{14} \oplus T_{62},
\]

\[
\Theta(c_{22}, Y_2, c_{21}, Y_3) \oplus T_{11},
\]

\[
\Theta(W_6, W_4, W_5, 0) \oplus T_{12},
\]

\[
\Theta(c_{23}, Y_1, c_{20}, Y_0) \oplus T_{13} \oplus T_{15},
\]

\[
\Theta(c_{24}, \Theta(Y_0, c_{23}, Y_1, c_{20}) \oplus T_0,
\]

\[
\Theta(c_{27}, \Theta(Y_3, c_{22}, Y_2, c_{21}) \oplus T_1) \oplus T_2,
\]

\[
\Theta(7S[W_4] \oplus 7S[W_5] \oplus 3S[W_6] \oplus T_6 \oplus T_{61},
\]

\[
\Theta(c_{20}, Y_0, c_{23}, Y_1) \oplus T_3 \oplus T_7)
\] \oplus T_{17} \oplus c_{33} = 0

3. Based on Equation (4.27):

\[
\Theta(\Theta(c_{26}, \Theta(Y_2, c_{21}, Y_3, c_{22}) \oplus T_8, c_{25}, \Theta(Y_1, c_{20}, Y_0, c_{23}) \oplus T_9) \oplus T_{18},
\]

\[
\Theta(7S[W_0] \oplus 7S[W_1] \oplus 3S[W_2] \oplus T_{11} \oplus T_{19},
\]

\[
\Theta(c_{22}, Y_2, c_{21}, Y_3) \oplus T_{14} \oplus T_{62},
\]

\[
\Theta(c_{27}, \Theta(Y_3, c_{22}, Y_2, c_{21}) \oplus T_1, c_{24}, \Theta(Y_0, c_{23}, Y_1, c_{20}) \oplus T_0) \oplus T_{20}
\]

\[
\Theta(7S[W_4] \oplus 7S[W_5] \oplus 3S[W_6] \oplus T_6 \oplus T_{61},
\]

\[
\Theta(c_{20}, Y_0, c_{23}, Y_1) \oplus T_3,
\]

\[
\Theta(W_2, W_0, W_1, 0) \oplus T_4,
\]

\[
\Theta(c_{21}, Y_3, c_{22}, Y_2) \oplus T_5,
\]

\[
\Theta(W_6, W_4, W_5, 0) \oplus T_{12},
\]

\[
\Theta(c_{23}, Y_1, c_{20}, Y_0) \oplus T_{13},
\]

\[
\Theta(c_{24}, \Theta(Y_0, c_{23}, Y_1, c_{20}) \oplus T_0) \oplus T_{20}
\]

\[
\Theta(c_{27}, \Theta(Y_3, c_{22}, Y_2, c_{21}) \oplus T_1, c_{24}, \Theta(Y_0, c_{23}, Y_1, c_{20}) \oplus T_0) \oplus T_{20}
\]

\[
\Theta(7S[W_4] \oplus 7S[W_5] \oplus 3S[W_6] \oplus T_6 \oplus T_{61},
\]

\[
\Theta(c_{20}, Y_0, c_{23}, Y_1) \oplus T_3,
\]

\[
\Theta(W_2, W_0, W_1, 0) \oplus T_4,
\]

\[
\Theta(c_{21}, Y_3, c_{22}, Y_2) \oplus T_5) \oplus T_{21})
\] \oplus T_{22} \oplus c_{34} = 0
4. Based on Equation (4.28):

\[
\Theta(\Theta( c_{27}, \Theta(Y_3, c_{22}, Y_2, c_{21}) \oplus T_1, \\
c_{24}, \Theta(Y_0, c_{23}, Y_1, c_{20}) \oplus T_0) \oplus T_{20}, \\
\Theta(7S[W_4] \oplus 7S[W_5] \oplus 3S[W_6] \oplus T_6 \oplus T_{61}, \\
\Theta(c_{20}, Y_0, c_{23}, Y_1) \oplus T_3, \\
\Theta(W_2, W_0, W_1, 0) \oplus T_4, \\
\Theta(c_{21}, Y_3, c_{22}, Y_2) \oplus T_5, \\
\Theta(c_{26}, \Theta(Y_2, c_{21}, Y_3, c_{22}) \oplus T_8, \\
c_{25}, \Theta(Y_1, c_{20}, Y_0, c_{23}) \oplus T_9) \oplus T_{18} \\
\Theta(W_6, W_4, W_5, 0) \oplus T_{12}, \\
\Theta(c_{23}, Y_1, c_{20}, Y_0) \oplus T_{13}, \\
7S[W_0] \oplus 7S[W_1] \oplus 3S[W_2] \oplus T_{14} \oplus T_{62}, \\
\Theta(c_{22}, Y_2, c_{21}, Y_3) \oplus T_{11}) \oplus T_{19}) \\
\oplus T_{23} \oplus c_{35} = 0
\]

5. Based on Equation (4.29):

\[
\hat{\Pi}(\hat{\Pi}( S[c_8], \hat{\Pi}(Y_6, S[c_{13}], Y_7, S[c_{14}]) \oplus T_{24}, \\
S[c_9], \hat{\Pi}(Y_5, S[c_{12}], Y_4, S[c_{13}]) \oplus T_{25}) \oplus T_{26}, \\
\hat{\Pi}(\hat{\Pi}(Z_{13}, Z_{14}, Z_{15}, Z_{12}) \oplus T_{28}, \\
\hat{\Pi}(S[c_{13}], Y_5, S[c_{12}], Y_4) \oplus T_{29}, \\
\hat{\Pi}(Z_{11}, Z_8, Z_9, Z_{10}) \oplus T_{30}, \\
\hat{\Pi}(S[c_{14}], Y_6, S[c_{15}], Y_7) \oplus T_{27}) \oplus T_{31}, \\
\hat{\Pi}(S[c_{11}], \hat{\Pi}(Y_7, S[c_{14}], Y_6, S[c_{15}]) \oplus T_{33}, \\
S[c_{10}], \hat{\Pi}(Y_4, S[c_{13}], Y_5, S[c_{12}]) \oplus T_{32}) \oplus T_{34}, \\
\hat{\Pi}(\hat{\Pi}(Z_{15}, Z_{12}, Z_{13}, Z_{14}) \oplus T_{38}, \\
\hat{\Pi}(S[c_{12}], Y_4, S[c_{13}], Y_5) \oplus T_{35}, \\
\hat{\Pi}(Z_9, Z_{10}, Z_{11}, Z_8) \oplus T_{36}, \\
\hat{\Pi}(S[c_{15}], Y_7, S[c_{14}], Y_6) \oplus T_{37}) \oplus T_{39}) \\
\oplus T_{40} \oplus S[c_0] = 0
\]
6. Based on Equation (4.30):

\[ \hat{\Pi}(\hat{\Pi}(S[c_9], \hat{\Pi}(Y_5, S[c_{12}], Y_4, S[c_{13}]) \oplus T_{25}, \]
\[ S[c_8], \hat{\Pi}(Y_6, S[c_{15}], Y_7, S[c_{14}]) \oplus T_{24}) \oplus T_{43}, \]
\[ \hat{\Pi}(Z_{11}, Z_8, Z_9, Z_{10}) \oplus T_{30}, \]
\[ \hat{\Pi}(S[c_{14}], Y_6, S[c_{15}], Y_7) \oplus T_{27}, \]
\[ \hat{\Pi}(Z_{13}, Z_{14}, Z_{15}, Z_{12}) \oplus T_{28}, \]
\[ \hat{\Pi}(S[c_{13}], Y_5, S[c_{12}], Y_4) \oplus T_{29}) \oplus T_{44}, \]
\[ \hat{\Pi}(S[c_{10}], \hat{\Pi}(Y_4, S[c_{13}], Y_5, S[c_{12}]) \oplus T_{32}, \]
\[ S[c_{11}], \hat{\Pi}(Y_7, S[c_{14}], Y_6, S[c_{15}]) \oplus T_{33}) \oplus T_{41}, \]
\[ \hat{\Pi}(Z_9, Z_{10}, Z_{11}, Z_8) \oplus T_{36}, \]
\[ \hat{\Pi}(S[c_{15}], Y_7, S[c_{14}], Y_6) \oplus T_{37}, \]
\[ \hat{\Pi}(Z_{15}, Z_{12}, Z_{13}, Z_{14}) \oplus T_{38}, \]
\[ \hat{\Pi}(S[c_{12}], Y_4, S[c_{13}], Y_5) \oplus T_{35}) \oplus T_{42}) \]
\[ \oplus T_{45} \oplus S[c_1] = 0 \]

7. Based on Equation (4.31):

\[ \hat{\Pi}(\hat{\Pi}(S[c_{10}], \hat{\Pi}(Y_4, S[c_{13}], Y_5, S[c_{12}]) \oplus T_{32}, \]
\[ S[c_{11}], \hat{\Pi}(Y_7, S[c_{14}], Y_6, S[c_{15}]) \oplus T_{33}) \oplus T_{41}, \]
\[ \hat{\Pi}(Z_9, Z_{10}, Z_{11}, Z_8) \oplus T_{36}, \]
\[ \hat{\Pi}(S[c_{15}], Y_7, S[c_{14}], Y_6) \oplus T_{37}, \]
\[ \hat{\Pi}(Z_{15}, Z_{12}, Z_{13}, Z_{14}) \oplus T_{38}, \]
\[ \hat{\Pi}(S[c_{12}], Y_4, S[c_{13}], Y_5) \oplus T_{35}, \]
\[ \hat{\Pi}(S[c_9], \hat{\Pi}(Y_5, S[c_{12}], Y_4, S[c_{13}]) \oplus T_{25}, \]
\[ S[c_8], \hat{\Pi}(Y_6, S[c_{15}], Y_7, S[c_{14}]) \oplus T_{24}) \oplus T_{43} \]
\[ \hat{\Pi}(Z_{11}, Z_8, Z_9, Z_{10}) \oplus T_{30}, \]
\[ \hat{\Pi}(S[c_{14}], Y_6, S[c_{13}], Y_7) \oplus T_{27}, \]
\[ \hat{\Pi}(Z_{13}, Z_{14}, Z_{15}, Z_{12}) \oplus T_{28}, \]
\[ \hat{\Pi}(S[c_{13}], Y_5, S[c_{12}], Y_4) \oplus T_{29}) \oplus T_{44}) \]
\[ \oplus T_{46} \oplus S[c_2] = 0 \]
8. Based on Equation (4.32):

\[
\hat{\Pi}(\hat{\Pi}(S[c_{11}], \hat{\Pi}(Y_7, S[c_{14}], Y_6, S[c_{15}]) \oplus T_{33},
S[c_{10}], \hat{\Pi}(Y_4, S[c_{13}], Y_5, S[c_{12}]) \oplus T_{32}) \oplus T_{34},
\hat{\Pi}(Z_{15}, Z_{12}, Z_{13}, Z_{14}) \oplus T_{38},
\hat{\Pi}(S[c_{12}], Y_4, S[c_{13}], Y_5) \oplus T_{35},
\hat{\Pi}(Z_{9}, Z_{10}, Z_{11}, Z_{8}) \oplus T_{36},
\hat{\Pi}(S[c_{15}], Y_7, S[c_{14}], Y_6) \oplus T_{37}) \oplus T_{39},
\hat{\Pi}(S[c_{8}], \hat{\Pi}(Y_6, S[c_{15}], Y_7, S[c_{14}]) \oplus T_{24},
S[c_{9}], \hat{\Pi}(Y_5, S[c_{12}], Y_4, S[c_{13}]) \oplus T_{25}) \oplus T_{26},
\hat{\Pi}(Z_{13}, Z_{14}, Z_{15}, Z_{12}) \oplus T_{28},
\hat{\Pi}(S[c_{13}], Y_5, S[c_{12}], Y_4) \oplus T_{29},
\hat{\Pi}(Z_{11}, Z_{8}, Z_{9}, Z_{10}) \oplus T_{30},
\hat{\Pi}(S[c_{14}], Y_6, S[c_{15}], Y_7) \oplus T_{27}) \oplus T_{31})
\oplus T_{47} \oplus S[c_{3}] = 0
\]

9. Based on Equation (4.21):

\[
\Theta(\Theta(c_{20}, Y_0, c_{23}, Y_1) \oplus T_3, \Theta(W_2, W_0, W_1, 0) \oplus T_4,
\Theta(c_{21}, Y_3, c_{22}, Y_2) \oplus T_5,
7S[W_4] \oplus 7S[W_5] \oplus 3S[W_6] \oplus T_6 \oplus T_{61}
\oplus T_{48} \oplus c_{28} = 0
\]

10. Based on Equation (4.22):

\[
\Theta(\Theta(c_{21}, Y_3, c_{22}, Y_2) \oplus T_5,
7S[W_4] \oplus 7S[W_3] \oplus 3S[W_6] \oplus T_6 \oplus T_{61},
\Theta(c_{20}, Y_0, c_{23}, Y_1) \oplus T_3, \Theta(W_2, W_0, W_1, 0) \oplus T_4)
\oplus T_{49} \oplus c_{29} = 0
\]
11. Based on Equation (4.23):

\[
\begin{align*}
\Theta(\Theta(c_{22}, Y_2, c_{21}, Y_3) \oplus T_{11}, \Theta(W_6, W_4, W_5, 0) \oplus T_{12}, \\
\Theta(c_{23}, Y_1, c_{20}, Y_0) \oplus T_{13}, \\
7S[W_0] \oplus 7S[W_1] \oplus 3S[W_2] \oplus T_{14} \oplus T_{62}, \\
\oplus T_{50} \oplus c_{30} = 0
\end{align*}
\]

12. Based on Equation (4.24):

\[
\begin{align*}
\Theta(\Theta(c_{23}, Y_1, c_{20}, Y_0) \oplus T_{13}, \\
7S[W_0] \oplus 7S[W_1] \oplus 3S[W_2] \oplus T_{14} \oplus T_{62}, \\
\Theta(c_{22}, Y_2, c_{21}, Y_3) \oplus T_{11}, \Theta(W_6, W_4, W_5, 0) \oplus T_{12}) \\
\oplus T_{51} \oplus c_{31} = 0
\end{align*}
\]

13. Based on Equation (4.5):

\[
\begin{align*}
\hat{\Pi}(\hat{\Pi}(S[c_{12}], Y_4, S[c_{13}], Y_5) \oplus T_{35}, \\
\hat{\Pi}(Z_9, Z_{10}, Z_{11}, Z_8) \oplus T_{36}, \\
\hat{\Pi}(S[c_{15}], Y_7, S[c_{14}], Y_6) \oplus T_{37}, \\
\hat{\Pi}(Z_{15}, Z_{12}, Z_{13}, Z_{14}) \oplus T_{38} \oplus T_{52} \oplus S[c_{12}] = 0
\end{align*}
\]

14. Based on Equation (4.6):

\[
\begin{align*}
\hat{\Pi}(\hat{\Pi}(S[c_{13}], Y_5, S[c_{12}], Y_4) \oplus T_{29}, \\
\hat{\Pi}(Z_{11}, Z_8, Z_9, Z_{10}) \oplus T_{30}, \\
\hat{\Pi}(S[c_{14}], Y_6, S[c_{15}], Y_7) \oplus T_{27}, \\
\hat{\Pi}(Z_{13}, Z_{14}, Z_{15}, Z_{12}) \oplus T_{28} \oplus T_{53} \oplus S[c_{13}] = 0
\end{align*}
\]

15. Based on Equation (4.7):

\[
\begin{align*}
\hat{\Pi}(\hat{\Pi}(S[c_{14}], Y_6, S[c_{15}], Y_7) \oplus T_{27}, \\
\hat{\Pi}(Z_{13}, Z_{14}, Z_{15}, Z_{12}) \oplus T_{28}, \\
\hat{\Pi}(S[c_{13}], Y_5, S[c_{12}], Y_4) \oplus T_{29}, \\
\hat{\Pi}(Z_{11}, Z_8, Z_9, Z_{10}) \oplus T_{30} \oplus T_{54} \oplus S[c_{14}] = 0
\end{align*}
\]
16. Based on Equation (4.8):
\[ \hat{\Pi}(\hat{\Pi}(S_{c_{15}}, Y_7, S_{c_{14}}, Y_6) \oplus T_{37}, \hat{\Pi}(Z_{15}, Z_{12}, Z_{13}, Z_{14}) \oplus T_{38}, \hat{\Pi}(S_{c_{12}}, Y_4, S_{c_{13}}, Y_5) \oplus T_{35}, \hat{\Pi}(Z_9, Z_{10}, Z_{11}, Z_8) \oplus T_{36}) \oplus T_{55} \oplus S_{c_{15}} = 0 \]

17. Based on Equation (4.20):
\[ 7S[W_0] \oplus 4S[W_1] \oplus S[W_2] \oplus T_{56} \oplus c_{27} = 0 \]

18. Based on Equation (4.37):
\[ \hat{\Pi}(Z_8, Z_9, Z_{10}, Z_{11}) \oplus T_{57} \oplus S_{c_8} = 0 \]

\[ \hat{\Pi}(Z_{14}, Z_{15}, Z_{12}, Z_{13}) \oplus T_{58} \oplus S_{c_9} = 0 \]

20. Based on Equation (4.39):
\[ \hat{\Pi}(Z_{12}, Z_{13}, Z_{14}, Z_{15}) \oplus T_{59} \oplus S_{c_{10}} = 0 \]

\[ \hat{\Pi}(Z_{10}, Z_{11}, Z_8, Z_9) \oplus T_{60} \oplus S_{c_{11}} = 0 \]

The \( W_j, Y_j, Z_j \) and \( T_j \) expressions in the above equations are composed of the sum of the remaining state, key variables and constants.

The \( W_j, Y_j, Z_j \) and \( T_j \) expressions in the above equations are defined as follows. Replace subkey variable \( k_{56} \) with the following equation: \( k_{56} = 4S^{-1}[y_0 \oplus y_1 \oplus y_2 \oplus K_2 \oplus c_{25}] \oplus 4y_3 \) where \( y_i \) and \( K_2 \) are defined in Appendix C.1.4.

- \( W_0 = 2S_{c_{18}} \oplus 3S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S_{c_{14}} \oplus C9S_{c_{13}}] \oplus S_{c_{17}} \oplus S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S_{c_{12}} \oplus 44S_{c_{15}}] \oplus k_{48} \oplus k_{52} \oplus k_{56} \oplus S[k_{61}] \oplus RC^4 \)
- \( W_1 = 44S[x_8] \oplus C9S[x_4] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47k_{56} \oplus \)
$k_{57} \oplus CBk_{58} \oplus 47k_{60} \oplus k_{61} \oplus 47S[k_{61}] \oplus CBk_{62} \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{22} \oplus CBC_{23}$

- $W_3 = S[c_{16}] \oplus S[47x_2] \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus 2S[c_{19}] \oplus 3S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{50} \oplus S[k_{63}]$

- $W_4 = 2S[c_{16}] \oplus 3S[47x_2] \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4$

- $W_5 = 44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBC_{21}$

- $W_6 = S[c_{18}] \oplus S[47x_8] \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}]$

- $Y_0 = S[c_{18}] \oplus 2S[47x_8] \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 3S[c_{17}] \oplus S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{49} \oplus k_{53} \oplus k_{57} \oplus S[k_{62}]$

- $Y_1 = 3S[c_{16}] \oplus S[47x_2] \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{16}] \oplus 2S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus CBk_{50} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus CBS[k_{61}] \oplus 44S[c_{13}] \oplus CBRC^4 \oplus CBc_{20} \oplus 47c_{21} \oplus S^{-1}[2S[c_{18}] \oplus 3S[47x_8] \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{48} \oplus k_{50} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47k_{56} \oplus k_{57} \oplus CBk_{58} \oplus 47k_{60} \oplus k_{61} \oplus 47S[k_{61}] \oplus CBk_{62} \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{22} \oplus CBC_{23} \oplus S[S[c_{16}] \oplus S[47x_2] \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus 3S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{49} \oplus k_{50} \oplus S[k_{63}] \oplus RC^4 \oplus RC^5 \oplus c_{26}$

- $Y_2 = S[c_{16}] \oplus 2S[47x_2] \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus 3S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{49} \oplus S[k_{62}]$

- $Y_4 = 9c_{16} \oplus ES^{-1}[47S[x_8] \oplus CBS[x_4] \oplus 46k_{48} \oplus CBk_{50} \oplus 46k_{52} \oplus CBk_{54} \oplus 46k_{56} \oplus CBk_{58} \oplus 46k_{60} \oplus 46S[k_{61}] \oplus CBk_{62} \oplus 46RC^4 \oplus CBS[k_{63}] \oplus 46c_{22} \oplus CBC_{23}] \oplus$
\[ Y_5 = Bc_{18} \oplus DS^{-1}[47S[x_2] \oplus CBS[x_10] \oplus 46k_{48} \oplus CBk_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[k_{61}] \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{20} \oplus 46c_{21}] \oplus D2S[x_2] \oplus 59S[x_{10}] \oplus C1k_{48} \oplus Ec_{49} \oplus 45k_{50} \oplus CBk_{52} \oplus 4Ekc_{54} \oplus Dc_{55} \oplus DS[k_{60}] \oplus CBS[k_{61}] \oplus 4ES[k_{63}] \oplus CBRC^4 \oplus CBc_{20} \oplus 4Ec_{21} \oplus DS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{20}] \]

\[ Y_6 = 9c_{18} \oplus ES^{-1}[47S[x_2] \oplus CBS[x_10] \oplus 46k_{48} \oplus CBk_{50} \oplus CBk_{52} \oplus CBk_{54} \oplus 46S[k_{61}] \oplus CBS[k_{63}] \oplus 46RC^4 \oplus 46c_{26} \oplus CBc_{21}] \oplus Bc_{19} \oplus DS^{-1}[CBS[x_2] \oplus 47S[x_4] \oplus CBk_{48} \oplus 46k_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBk_{56} \oplus 46k_{58} \oplus CBk_{60} \oplus CBS[k_{61}] \oplus 46k_{62} \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{22} \oplus CBc_{24}] \oplus D2S[x_2] \oplus 59S[x_4] \oplus 59S[x_{10}] \oplus D2S[x_8] \oplus C1k_{56} \oplus Ec_{57} \oplus 45k_{55} \oplus C8k_{60} \oplus 4Ec_{62} \oplus Dc_{63} \oplus CBc_{26} \oplus 4Ec_{21} \oplus CBc_{22} \oplus 4Ec_{23} \oplus DS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{20}] \]

\[ Y_7 = Bc_{16} \oplus DS^{-1}[47S[x_8] \oplus CBS[x_4] \oplus 46k_{48} \oplus CBk_{50} \oplus CBk_{52} \oplus CBk_{54} \oplus 46k_{56} \oplus 46k_{58} \oplus CBk_{60} \oplus CBS[k_{61}] \oplus 46k_{62} \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{20} \oplus 46c_{21}] \oplus D2S[x_2] \oplus 59S[x_{10}] \oplus C1k_{48} \oplus Ec_{49} \oplus 45k_{50} \oplus CBk_{52} \oplus 4Ekc_{54} \oplus Dc_{55} \oplus DS[k_{60}] \oplus CBS[k_{61}] \oplus 4ES[k_{63}] \oplus CBRC^4 \oplus CBc_{20} \oplus 4Ec_{21} \oplus DS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{20}] \]
C.1. Equations

\[ C_{15}^{58} \oplus 46k_{60} \oplus 46S[k_{61}] \oplus CBk_{62} \oplus CBS[k_{63}] \oplus 46RC^4 \oplus 46c_{22} \oplus CBc_{23} \oplus 9c_{17} \oplus ES^{-1}[CBS[x_2] \oplus 47S[x_{10}] \oplus CBk_{48} \oplus 46k_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[k_{61}] \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{20} \oplus 46c_{21} \oplus 92S[x_2] \oplus 95S[x_{10}] \oplus 85k_{49} \oplus k_{49} \oplus 8Ek_{50} \oplus 8Ek_{52} \oplus 87k_{54} \oplus ES[k_{60}] \oplus 8ES[k_{61}] \oplus 87S[k_{63}] \oplus 8ERC^4 \oplus 8Ec_{20} \oplus 87c_{21} \oplus ES^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{20}] \]

- \[ Z_8 = E_{c_{16}} \oplus BS^{-1}[47S[x_8] \oplus CBS[x_4] \oplus 46k_{48} \oplus CBk_{50} \oplus 46k_{52} \oplus CBk_{54} \oplus 46k_{56} \oplus CBk_{58} \oplus 46k_{60} \oplus 46S[k_{61}] \oplus CBk_{62} \oplus CBS[k_{63}] \oplus 46RC^4 \oplus 46c_{22} \oplus CBc_{23} \oplus DC_{17} \oplus 9S^{-1}[CBS[x_2] \oplus 47S[x_{10}] \oplus CBk_{48} \oplus 46k_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[k_{61}] \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{20} \oplus 46c_{21} \oplus DBS[x_2] \oplus 52S[x_{10}] \oplus 7c_{48} \oplus Bk_{49} \oplus 44k_{50} \oplus C9k_{52} \oplus 49k_{54} \oplus 9k_{55} \oplus 9S[k_{60}] \oplus C9S[k_{61}] \oplus 49S[k_{63}] \oplus C9RC^4 \oplus C9c_{20} \oplus 49c_{21} \oplus 9S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{20}] \]

- \[ Z_0 = 46x_8 \oplus CBx_{10} \oplus 46k_{60} \oplus CBk_{62} \oplus 47c_{14} \oplus CBS[c_{13}] \]

- \[ Z_{10} = DC_{18} \oplus 9S^{-1}[47S[x_{2}] \oplus CBS[x_{10}] \oplus 46k_{48} \oplus CBk_{50} \oplus 46k_{52} \oplus CBk_{54} \oplus 46S[k_{61}] \oplus CBS[k_{63}] \oplus 46RC^4 \oplus 46c_{20} \oplus CBc_{21} \oplus E_{c_{19}} \oplus BS^{-1}[CBS[x_{8}] \oplus 47S[x_{4}] \oplus CBk_{48} \oplus 46k_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[k_{61}] \oplus 46k_{62} \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{22} \oplus 46c_{23} \oplus 52S[x_{2}] \oplus DAS[x_{4}] \oplus DAS[x_{10}] \oplus 52S[x_{8}] \oplus 49k_{56} \oplus 9k_{57} \oplus C9k_{58} \oplus 44k_{60} \oplus 7c_{62} \oplus Bk_{63} \oplus 44c_{20} \oplus 7c_{21} \oplus 44c_{22} \oplus C7c_{23} \oplus BS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \]

- \[ BS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \]

- \[ BS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \]

- \[ Z_{11} = CBx_{2} \oplus 46x_{4} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[c_{12}] \oplus 47S[c_{15}] \]

- \[ Z_{12} = E_{c_{18}} \oplus BS^{-1}[47S[x_2] \oplus CBS[x_{10}] \oplus 46k_{48} \oplus CBk_{50} \oplus 46k_{52} \oplus CBk_{54} \oplus 46S[k_{61}] \oplus CBS[k_{63}] \oplus 46RC^4 \oplus 46c_{20} \oplus CBc_{21} \oplus DC_{19} \oplus 9S^{-1}[CBS[x_8] \oplus 47S[x_4] \oplus CBk_{48} \oplus 46k_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[k_{61}] \oplus 46k_{62} \oplus 46S[k_{63}] \oplus CBRC^4 \oplus CBc_{22} \oplus 46c_{23} \oplus DBS[x_2] \oplus 52S[x_4] \oplus 52S[x_{10}] \oplus DBS[x_8] \oplus \ldots \]
\[ C7k_{56} \oplus Bk_{57} \oplus 44k_{58} \oplus C9k_{60} \oplus 49k_{62} \oplus 9k_{63} \oplus C9c_{20} \oplus 49c_{21} \oplus C9c_{22} \oplus 49c_{23} \oplus 9S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{60} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus 9S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{21}] \]

- \[ Z_{13} = 46x_2 \oplus CBx_4 \oplus 46k_{52} \oplus CBk_{54} \oplus 47S[c_{12}] \oplus CBS[c_{15}] \]
- \[ Z_{14} = DC_{16} \oplus 9S^{-1}[47S[x_8] \oplus CBS[x_4] \oplus 46k_{48} \oplus CBk_{50} \oplus 46k_{52} \oplus CBk_{54} \oplus 46k_{56} \oplus CBk_{58} \oplus 46k_{60} \oplus 46S[k_{61}] \oplus CBk_{62} \oplus 46RC^4 \oplus CBS[k_{63}] \oplus 46c_{22} \oplus CBc_{23}] \oplus Ec_{17} \oplus BS^{-1}[CBS[x_2] \oplus 47S[x_{10}] \oplus CBk_{48} \oplus 46k_{50} \oplus CBk_{52} \oplus 46k_{54} \oplus CBS[k_{61}] \oplus 46S[k_{63}] \oplus CBS[k_{64}] \oplus CBS[k_{65}] \oplus BCRC^4 \oplus CBx_{20} \oplus 46c_{21} \oplus 52S[x_4] \oplus DA5S[x_{10}] \oplus 49k_{48} \oplus 9k_{49} \oplus C9k_{50} \oplus 44k_{52} \oplus C7k_{54} \oplus Bk_{55} \oplus BS[k_{60}] \oplus 44S[k_{61}] \oplus C7S[k_{63}] \oplus 44RC^4 \oplus 44c_{20} \oplus 47c_{21} \oplus BS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{20}] \]

- \[ Z_{15} = CBx_8 \oplus 46x_{10} \oplus CBk_{60} \oplus 46k_{62} \oplus CBS[c_{14}] \oplus 47S[c_{13}] \]

- \[ T_0 = k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}] \]

- \[ T_1 = k_{55} \oplus k_{63} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \]

- \[ T_2 = k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus S[k_{61} \oplus k_{53} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}] \oplus RC^4 \oplus RC^5 \oplus RC^6 \]

- \[ T_3 = k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5 \]

- \[ T_4 = k_{48} \oplus k_{49} \oplus k_{56} \oplus k_{57} \oplus S[k_{61}] \oplus S[k_{62}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \]

- \[ T_5 = k_{54} \oplus k_{62} \oplus S[44S[x_4] \oplus C9S[x_8] \oplus CBk_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus CBS[k_{61}] \oplus 47k_{62} \oplus 47S[k_{63}] \oplus CBS[k_{63}] \oplus CBRC^4 \oplus CBS[C22] \oplus 47c_{23}] \oplus S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{16}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus
C.1. Equations

\[
CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_{8} \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_{2} \oplus 47x_{4} \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]
\]

- \( T_0 = C9S[x_{2}] \oplus 44S[x_{10}] \oplus C9k_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[k_{61}] \oplus 47S[k_{63}] \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 2S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{19} \oplus S[k_{62}] \oplus C9RC^4 \oplus 2RC^5 \oplus CBc_{20} \oplus 47c_{21} \oplus 2c_{24} \)

- \( T_7 = k_{49} \oplus k_{53} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{54} \oplus S[44S[x_{4}] \oplus C9S[x_{8}] \oplus CBk_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus CBS[k_{61}] \oplus 47k_{62} \oplus 47S[k_{63}] \oplus CBRC^4 \oplus CBc_{22} \oplus 47c_{23} \oplus S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_{2} \oplus CBx_{4} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_{8} \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{16} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_{2}] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_{8} \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_{2} \oplus 47x_{4} \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]]
\)

- \( T_8 = k_{53} \oplus k_{61} \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}] \)

- \( T_9 = k_{55} \oplus S[k_{60} \oplus k_{56} \oplus k_{52} \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \)

- \( T_{10} = k_{54} \oplus k_{58} \oplus S[44S[x_{4}] \oplus C9S[x_{8}] \oplus CBk_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus CBS[k_{61}] \oplus 47k_{62} \oplus 47S[k_{63}] \oplus CBRC^4 \oplus CBc_{22} \oplus 47c_{23} \oplus S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_{2} \oplus CBx_{4} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_{8} \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_{2}] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_{8} \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_{2} \oplus 47x_{4} \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \oplus S[k_{63}] \oplus T_9 \]

- \( T_{11} = k_{52} \oplus k_{60} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5 \)

- \( T_{12} = k_{48} \oplus k_{49} \oplus S[k_{61}] \oplus S[k_{62}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus S[k_{62} \oplus k_{58} \oplus k_{54} \oplus k_{50} \oplus S[k_{63}] \oplus RC^4 \oplus RC^5 \oplus c_{24} \)

- \( T_{13} = k_{54} \oplus S[44S[x_{4}] \oplus C9S[x_{8}] \oplus CBk_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus CBS[k_{61}] \oplus 47k_{62} \oplus 47S[k_{63}] \oplus CBRC^4 \oplus CBc_{22} \oplus 47c_{23} \oplus \)


$S^{-1}[2S[2S[c_{16}]] + 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{16}]]) + S[c_{19}] + S[CBx_a \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}]]) + k_{48} + S[k_{61}] + RC^4] + 3S[44S[x_2] \oplus C9S[x_{10}] + 47k_{48} \oplus k_{49} \oplus CBk_{50} + 47k_{52} \oplus k_{53} \oplus CBk_{54} + 47S[k_{61}] + S[k_{62}] + C9S[k_{63}] + 47RC^4 + 47c_{20} + Cc_{21} + S[c_{18}] + S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}]]) + 2S[c_{17}] + 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}]]) + k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] + k_{48} + S[k_{61}] + S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + RC^4 + RC^5 + c_{24}]$

- $T_{14} = 44S[x_2] + C9S[x_8] + C9k_{48} + 47k_{50} + CBk_{52} + 47k_{54} + k_{55} + C9k_{56} + 47k_{58} + CBk_{60} + C9S[k_{61}] + 47k_{62} + 47S[k_{63}] + k_{63} + S[k_{60} + k_{56} + k_{52} + k_{48} + S[k_{61}] + RC^4] + 2S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + C9RC^4 + 2RC^5 + CBc_{22} + 47c_{23} + 2c_{26}$

- $T_{15} = C9S[x_2] + 44S[x_4] + 44S[x_{10}] + C9S[x_8] + CBk_{56} + 47k_{58} + CBk_{60} + 47k_{62} + CBc_{20} + 47c_{21} + CBc_{22} + 47c_{23} + S^{-1}[2S[W_0] + 3S[W_1] + S[W_2] + k_{48} \oplus k_{56} \oplus S[k_{61}] + S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + RC^4 + RC^5 + c_{26}] + S^{-1}[2S[2S[c_{16}]] + 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} + k_{53} + CBk_{54} + 44S[c_{12}] + C9S[c_{15}]) + S[c_{19}] + S[CBx_2 \oplus 47x_{10} \oplus CBk_{60} + 47k_{62} \oplus k_{63} + C9S[c_{14}] + 44S[c_{13}]]) + k_{48} + S[k_{61}] + RC^4] + 3S[44S[x_2] + C9S[x_{10}] + 47k_{48} \oplus k_{49} \oplus CBk_{50} + 47k_{52} \oplus k_{53} \oplus CBk_{54} + 47S[k_{61}] + S[k_{62}] + CB[k_{63}] + 47RC^4 + 47c_{20} + CBc_{21} + S[c_{18}] + S[47x_8 \oplus CBx_{10} \oplus 47k_{60} + k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] + C9S[c_{13}]) + 2S[c_{17}] + 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} + C9S[c_{12}] + 44S[c_{15}]]) + k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] + k_{48} + S[k_{61}] + S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + RC^4 + RC^5 + c_{24}] + S[k_{60} + k_{52} + S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + RC^5]$

- $T_{16} = k_{48} \oplus S[k_{61}] + S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + S[k_{61} + T_0] + S[k_{61} + k_{57} + S[k_{62} + T_{13}]] + RC^4 + RC^5 + RC^6 + RC^7$

- $T_{17} = T_{20} + S[C9S[x_2] + 44S[x_4] + 44S[x_{10}] + C9S[x_8] + CBk_{56} + 47k_{58} + CBk_{60} + 47k_{62} + k_{63} + k_{63} + S[k_{60} + T_3] + CBc_{20} + 47c_{21} + CBc_{22} + 47c_{23} + T_{61} + T_{62}$

- $T_{18} = k_{52} \oplus k_{56} \oplus S[k_{61} + k_{57} + k_{53} + k_{49} + S[k_{62}] + S[k_{61} + k_{53} + S[k_{62} + k_{58} + k_{54} + k_{50} + S[k_{63}]]) + RC^5 + RC^6$

- $T_{19} = k_{57} + k_{61} + S[k_{62} + T_{13}]$

- $T_{20} = k_{50} + S[k_{63}] + S[44S[x_4] + C9S[x_8] + CBk_{48} + 47k_{50} + CBk_{52} + 47k_{54} + CBk_{56} + 47k_{58} + CBk_{60} + CB[k_{61}] + 47k_{62} + 47S[k_{63}] + CBRC^4 + CBc_{22} + 47c_{23} + S^{-1}[2S[2S[c_{16}]] + 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} + k_{53} + CBk_{54} + 44S[c_{12}] + C9S[c_{15}]) + S[c_{19}] + S[CBx_2 \oplus 47x_{10} \oplus CBk_{60} + 47k_{62} + k_{63} + C9S[c_{14}] + 44S[c_{13}]) + k_{48}$
$S[k_{61}] + RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S][c_{18}] \oplus S[47x_2 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBc_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{18} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \oplus S[k_{63} \oplus T_9]$

- $T_{21} = C9S[x_2] \oplus 44S[x_{10}] \oplus CBk_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus CBS[k_{61}] \oplus 47S[k_{63}] \oplus S[k_{60} \oplus k_{52} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^5] \oplus CBRC^4 \oplus CBc_{20} \oplus 47c_{21} \oplus S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}]$

- $T_{22} = k_{56} \oplus S[k_{61} \oplus T_0] \oplus S[k_{61} \oplus k_{57} \oplus S[k_{62} \oplus T_{13}]] \oplus RC^6 \oplus RC^7$

- $T_{23} = k_{58} \oplus S[k_{63} \oplus T_9] \oplus S[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_{8}] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus k_{63} \oplus S[k_{60} \oplus T_3] \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62}$

- $T_{24} = D2S[x_2] \oplus 59S[x_4] \oplus 59S[x_{10}] \oplus D2S[x_8] \oplus C1k_{56} \oplus E4k_{57} \oplus 45k_{58} \oplus C1k_{60} \oplus E4k_{61} \oplus 45k_{62} \oplus C8c_{20} \oplus 4E_{c_21} \oplus C8c_{22} \oplus 4E_{c_23} \oplus DS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \oplus DS^{-1}[2S[2S[c_{16}]] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus CBk_{63} \oplus 47k_{65} \oplus k_{57} \oplus k_{53} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBk_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S][c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]$

- $T_{25} = 92S[x_2] \oplus 95S[x_{10}] \oplus 85k_{48} \oplus Dk_{49} \oplus 8Ek_{50} \oplus 85k_{52} \oplus Dk_{53} \oplus 8Ek_{54} \oplus ES[k_{60}] \oplus 8ES[k_{61}] \oplus 87S[k_{63}] \oplus 8ERC^4 \oplus 8c_{20} \oplus 87c_{21} \oplus ES^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}]$

- $T_{26} = DBS[x_2] \oplus 52S[x_{10}] \oplus 7Ck_{48} \oplus BK_{49} \oplus 44k_{50} \oplus C9k_{52} \oplus 49k_{54} \oplus 9k_{55} \oplus 9S[k_{60}] \oplus C9S[k_{61}] \oplus 49S[k_{63}] \oplus ES[k_{61} \oplus k_{57}] \oplus ES[k_{61} \oplus k_{53}] \oplus BS[k_{62} \oplus k_{58}] \oplus BS[k_{62} \oplus k_{54}] \oplus DS[k_{63} \oplus k_{55}] \oplus 9S[k_{60} \oplus k_{56}] \oplus 9S[k_{60} \oplus k_{52}] \oplus C9RC^4 \oplus ERC^2 \oplus ERC^3 \oplus C9c_{20} \oplus 49c_{21} \oplus 9S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \oplus DS[C9S[x_2] \oplus 44S[x_4] \oplus
Appendix C. LEX-AES Equations

\[
44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus k_{63} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus S^{-1}\{2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus S[k_{57}] \oplus S[k_{53}] \oplus S[k_{63}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus S^{-1}\{2S[S_{c_0} \oplus 3S[C9S[c_{02}] \oplus C9S[c_{03}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{60}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S_{c_1} \oplus 47x_{8} \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]
\]

- \( T_{27} = DBS[x_2] \oplus 52S[x_4] \oplus 52S[x_{10}] \oplus DBS[x_8] \oplus C7k_{56} \oplus Bk_{57} \oplus 44k_{58} \oplus C7k_{60} \oplus BK_{61} \oplus 49k_{62} \oplus Bk_{62} \oplus C9c_{20} \oplus 49c_{21} \oplus C9c_{22} \oplus 49c_{23} \oplus 9S^{-1}\{2S[W_0] \oplus S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus S[k_{57}] \oplus S[k_{53}] \oplus S[k_{59}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus 9S^{-1}\{2S[S_{c_0} \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus 9S[C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{60}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S_{c_1} \oplus 47x_{8} \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]
\]

- \( T_{28} = D2S[x_2] \oplus 59S[x_4] \oplus 59S[x_{10}] \oplus D2S[x_8] \oplus 9k_{52} \oplus Ek_{53} \oplus Bk_{54} \oplus Bk_{55} \oplus C1k_{56} \oplus Ek_{57} \oplus 45k_{58} \oplus CBk_{60} \oplus 4Ek_{62} \oplus Dk_{63} \oplus CBc_{20} \oplus 4Ec_{21} \oplus CBc_{22} \oplus 4Ec_{23} \oplus DS^{-1}\{2S[W_0] \oplus S[S_{c_1} \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus S[k_{57}] \oplus S[k_{53}] \oplus S[k_{49}] \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus DS^{-1}\{2S[S_{c_0} \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{60}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S_{c_1} \oplus 47x_{8} \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]
\]

- \( T_{29} = 52S[x_2] \oplus DAS[x_{10}] \oplus 49k_{48} \oplus 9k_{49} \oplus C9k_{50} \oplus 49k_{52} \oplus 9k_{53} \oplus C9k_{54} \oplus BS[k_{60}] \oplus 44S[k_{61}] \oplus C7s[k_{63}] \oplus 44RC^4 \oplus 44c_{20} \oplus C7c_{21} \oplus BS^{-1}\{2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]
\]
\[ RC^4 \oplus RC^5 \oplus c_{20} \]

- \( T_{30} = 92S[x_2] \oplus 95S[x_{10}] \oplus 85k_{48} \oplus Dk_{49} \oplus 8Ek_{50} \oplus 8Ek_{52} \oplus 87k_{54} \oplus Ek_{55} \oplus ES[k_{60}] \oplus 8ES[k_{61}] \oplus 87S[k_{63}] \oplus BS[k_{61} \oplus k_{57}] \oplus DS[k_{62} \oplus k_{58}] \oplus ES[k_{60} \oplus k_{56}] \oplus BRC^3 \oplus 8ERC^4 \oplus 8Ec_{20} \oplus 87c_{21} \oplus ES^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus 9S[C9S[x_2] \oplus 44S[x_1] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CB_{k_{36}} \oplus 47k_{58} \oplus CB_{k_{60}} \oplus 47k_{62} \oplus CB_{c_{20}} \oplus 47c_{21} \oplus CB_{c_{22}} \oplus 47c_{23} \oplus S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}]] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CB_{x_4} \oplus 47k_{52} \oplus k_{53} \oplus CB_{k_{54}} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CB_{x_8} \oplus 47x_{10} \oplus CB_{k_{60}} \oplus 47k_{62} \oplus CB_{k_{62}} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CB_{k_{50}} \oplus 47k_{52} \oplus k_{53} \oplus CB_{k_{54}} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CB_{k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CB_{c_{21}] \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CB_{x_{10}] \oplus 47k_{60} \oplus CB_{k_{62}} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CB_{x_2} \oplus 47x_4 \oplus CB_{k_{62}} \oplus 47k_{54} \oplus k_{53} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \]

- \( T_{31} = 9k_{52} \oplus Ek_{53} \oplus Bk_{54} \oplus Dk_{55} \oplus 9k_{60} \oplus Ek_{61} \oplus Bk_{62} \oplus Dk_{63} \)

- \( T_{32} = D2S[x_2] \oplus 59S[x_{10}] \oplus C1k_{48} \oplus EK_{49} \oplus 45k_{50} \oplus C1k_{52} \oplus Ek_{53} \oplus 45k_{54} \oplus DS[k_{60}] \oplus CS[k_{61}] \oplus 4ES[k_{63}] \oplus C8RC^4 \oplus C8c_{20} \oplus 4Ec_{21} \oplus DS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus ES^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CB_{x_4} \oplus 47k_{52} \oplus k_{53} \oplus CB_{k_{54}} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CB_{x_8} \oplus 47x_{10} \oplus CB_{k_{60}} \oplus 47k_{62} \oplus CB_{k_{62}} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CB_{k_{50}} \oplus 47k_{52} \oplus k_{53} \oplus CB_{k_{54}} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CB_{k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CB_{c_{21}] \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CB_{x_{10}] \oplus 47k_{60} \oplus CB_{k_{62}} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CB_{x_2} \oplus 47x_4 \oplus CB_{k_{62}} \oplus 47k_{54} \oplus k_{53} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \]

- \( T_{34} = DAS[x_2] \oplus 52S[x_8] \oplus 49k_{48} \oplus 9k_{49} \oplus C9k_{50} \oplus 44k_{52} \oplus 7k_{54} \oplus Bk_{55} \oplus 49k_{56} \oplus 9k_{57} \oplus C9k_{58} \oplus BS[k_{60}] \oplus 44k_{60} \oplus 44S[k_{61}] \oplus C7k_{62} \oplus 7S[k_{63}] \oplus Bk_{63} \oplus 44RC^4 \oplus 44c_{22} \oplus C7c_{23} \oplus BS^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CB_{x_4} \oplus 47k_{52} \oplus k_{53} \oplus CB_{k_{54}}} \]
\[44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4] \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{63}] \oplus CB[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24} \]

- \[T_{35} = DBS[x_2] \oplus 52S[x_{10}] \oplus 7k_{48} \oplus BK_{49} \oplus 44k_{50} \oplus 7k_{52} \oplus BK_{53} \oplus 44k_{54} \oplus 9S[k_{60}] \oplus C9S[k_{61}] \oplus 49S[k_{63}] \oplus C9RC^4 \oplus C9c_{20} \oplus 49c_{21} \oplus 9S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \]

- \[T_{36} = DS[x_2] \oplus 59S[x_{10}] \oplus C1k_{48} \oplus EIk_{49} \oplus 45k_{50} \oplus 8k_{52} \oplus 4Eik_{54} \oplus BK_{55} \oplus DS[k_{60}] \oplus CB8S[k_{61}] \oplus ES[k_{63}] \oplus C8RC^4 \oplus C8c_{20} \oplus 4Eck_{21} \oplus DS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus 9S[k_{61} \oplus k_{57}] \oplus ES[k_{62} \oplus k_{58}] \oplus BS[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus CBc_{22} \oplus 47c_{23} \oplus S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus 9S[k_{61} \oplus k_{57}] \oplus ES[k_{62} \oplus k_{58}] \oplus BS[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CB[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{21}] \oplus DS[k_{60} \oplus k_{56}] \oplus 9RC^3 \]

- \[T_{37} = 52S[x_2] \oplus DAS[x_4] \oplus DAS[x_{10}] \oplus 52S[x_8] \oplus 49k_{56} \oplus 9k_{57} \oplus CBk_{58} \oplus 52S[x_6] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus BS^{-1}[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_{4} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63}] \oplus 44S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus S[44S[x_2] \oplus C9S[c_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CB[k_{63}] \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \]

\[ T_{38} = 92S[x_2] \oplus 95S[x_4] \oplus 95S[x_{10}] \oplus 92S[x_8] \oplus Bk_{52} \oplus Dk_{53} \oplus 9k_{54} \oplus E_k{55} \oplus 85k_{56} \oplus Dk_{57} \oplus 8E_k{58} \oplus 8E_k{60} \oplus 87k_{62} \oplus E_k{63} \oplus 8E_k{20} \oplus 8E_k{22} \oplus 87c_{23} \oplus E^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{62}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \oplus E^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBS[k_{63}] \oplus 47RC^4 \oplus 47C_{20} \oplus CBC_{21} \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBS[k_{63}] \oplus BCk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \] \\

\[ T_{39} = Bk_{52} \oplus Dk_{53} \oplus 9k_{54} \oplus E_k{55} \oplus BS[k_{61} \oplus k_{57}] \oplus DS[k_{62} \oplus k_{58}] \oplus ES[k_{60} \oplus k_{56}] \oplus BR^C \oplus 9S[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus k_{63} \oplus CBC_{20} \oplus 47c_{21} \oplus CBC_{22} \oplus 47c_{23} \oplus S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{62}] \oplus k_{49} \oplus S[k_{63}] \oplus RC^4 \oplus RC^5 \oplus c_{26}] \oplus S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \] \\

\[ T_{40} = DBS[x_2] \oplus 52S[x_{10}] \oplus C7k_{18} \oplus Bk_{49} \oplus 44k_{50} \oplus 49k_{52} \oplus 49k_{54} \oplus 9k_{55} \oplus 9S[k_{60} \oplus C9S[k_{61}] \oplus 49S[k_{63}] \oplus DS[k_{63} \oplus k_{55}] \oplus ES[k_{61} \oplus k_{57}] \oplus ES[k_{61} \oplus k_{53}] \oplus ES[k_{61} \oplus k_{53}] \oplus k_{53} \oplus k_{57} \oplus k_{49}] \oplus BS[k_{62} \oplus k_{58}] \oplus BS[k_{62} \oplus k_{54} \oplus k_{58} \oplus k_{50}] \oplus 9S[k_{60} \oplus k_{56}] \oplus 9S[k_{60} \oplus k_{52}] \oplus 9S[k_{60} \oplus k_{52} \oplus k_{56} \oplus k_{49}] \oplus ERC^1 \oplus ERC^2 \oplus ERC^3 \oplus 9R^C \oplus 9C_{20} \oplus 49c_{21} \oplus 9T_{61} \oplus DS[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBC_{20} \oplus 47c_{21} \oplus CBC_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62} \oplus DS[44S[x_4] \oplus C9S[x_8] \oplus CBk_{48} \oplus 47k_{50} \oplus CBk_{52} \oplus 47k_{54} \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus S[k_{60}] \oplus CBS[k_{61}] \oplus 47S[k_{63}] \oplus 47k_{62} \oplus CBRC^4 \oplus CBC_{22} \oplus 47c_{23} \oplus T_{62} \] \\

\[ T_{41} = 52S[x_4] \oplus DBS[x_8] \oplus C7k_{48} \oplus Bk_{49} \oplus 44k_{50} \oplus 49k_{52} \oplus 49k_{54} \oplus 9k_{55} \oplus C7k_{56} \oplus Bk_{57} \oplus Dk_{58} \oplus 49k_{58} \oplus C9k_{60} \oplus 9S[k_{60}] \oplus C9S[k_{61}] \oplus 49k_{62} \oplus 9k_{63} \oplus 49S[k_{63}] \oplus 9CRC^4 \oplus 9C_{22} \oplus 49c_{23} \oplus 9S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61}] \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \]
\[ 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 RC^5 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBk_{63} \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}] \]

\[ T_{42} = 9k_{52} \oplus Ek_{53} \oplus Bk_{54} \oplus Dk_{55} \oplus 9S[k_{61} \oplus k_{57}] \oplus ES[k_{62} \oplus k_{58}] \oplus DS[k_{60} \oplus k_{56}] \oplus 9RC^3 \oplus BS[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{50} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus 47c_{21} \oplus CBC_{22} \oplus 47c_{23} \oplus S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus S^{-1}[2S[c_{10}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus CBx_8 \oplus 47x_{10} \oplus CBk_6 \oplus 47k_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{51}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61} \oplus S[k_{62}] \oplus CBk_{56} \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[c_{18}] \oplus 47x_8 \oplus CBx_{10} \oplus 47k_{40} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]] \]

\[ T_{43} = 52S[x_2] \oplus DAS[x_{10}] \oplus 49k_{48} \oplus 9k_{49} \oplus C9k_{50} \oplus 44k_{52} \oplus C7k_{54} \oplus Bk_{55} \oplus BS[k_{60}] \oplus 44S[k_{61}] \oplus C7S[k_{63}] \oplus S[k_{61} \oplus k_{57}] \oplus DS[k_{61} \oplus k_{53}] \oplus 9S[k_{62} \oplus k_{58}] \oplus 9S[k_{62} \oplus k_{54}] \oplus ES[k_{63} \oplus k_{55}] \oplus BS[k_{60} \oplus k_{56}] \oplus BS[k_{60} \oplus k_{52}] \oplus 44RC^4 \oplus 44c_{20} \oplus C7c_{21} \oplus DRC^2 \oplus DRC^3 \oplus BS^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus ES[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus S^{-1}[2S[W_0] \oplus 3S[W_1] \oplus S[W_2] \oplus k_{48} \oplus k_{56} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{26} \oplus S^{-1}[2S[2S[c_{16}] \oplus 3S[47x_2 \oplus CBx_4 \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 44S[c_{12}] \oplus C9S[c_{15}] \oplus S[c_{19}] \oplus S[CBx_8 \oplus 47x_{10}] \oplus CBk_{60} \oplus 47k_{62} \oplus k_{63} \oplus C9S[c_{14}] \oplus 44S[c_{13}] \oplus k_{48} \oplus S[k_{61}] \oplus RC^4 \oplus 3S[44S[x_2] \oplus C9S[x_{10}] \oplus 47k_{48} \oplus k_{49} \oplus CBk_{50} \oplus 47k_{52} \oplus k_{53} \oplus CBk_{54} \oplus 47S[k_{61}] \oplus S[k_{62}] \oplus CBk_{56} \oplus 47RC^4 \oplus 47c_{20} \oplus CBc_{21} \oplus S[S[c_{18}] \oplus S[47x_8 \oplus CBx_{10} \oplus 47k_{60} \oplus k_{61} \oplus CBk_{62} \oplus 44S[c_{14}] \oplus C9S[c_{13}] \oplus 2S[c_{17}] \oplus 3S[CBx_2 \oplus 47x_4 \oplus CBk_{52} \oplus 47k_{54} \oplus k_{55} \oplus C9S[c_{12}] \oplus 44S[c_{15}] \oplus k_{50} \oplus k_{54} \oplus k_{58} \oplus S[k_{63}] \oplus k_{48} \oplus S[k_{61}] \oplus S[k_{61} \oplus k_{57} \oplus k_{53} \oplus k_{49} \oplus S[k_{62}] \oplus RC^4 \oplus RC^5 \oplus c_{24}]] \]

\[ T_{44} = Bk_{52} \oplus Dk_{53} \oplus 9k_{54} \oplus Ek_{55} \oplus Bk_{60} \oplus Dk_{61} \oplus 9k_{62} \oplus Ek_{63} \]

\[ T_{45} = DAS[x_4] \oplus 52S[x_4] \oplus 49k_{48} \oplus 9k_{49} \oplus C9k_{50} \oplus 49k_{52} \oplus 9k_{53} \oplus C9k_{54} \oplus \]
C.1. Equations

\[ 49k_{56} \oplus 9k_{57} \oplus C9k_{58} \oplus 44k_{60} \oplus BS[k_{60}] \oplus 44S[k_{61}] \oplus C7k_{62} \oplus C7S[k_{63}] \oplus Bk_{63} \oplus DS[k_{61} \oplus k_{57}] \oplus 9S[k_{62} \oplus k_{58}] \oplus BS[k_{60} \oplus k_{56}] \oplus DRC^3 \oplus 44RC^4 \oplus 44c_{22} \oplus C7c_{23} \oplus BT_{62} \oplus ES[C9S[x_1] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62} ] \]

- \[ T_{46} = 52S[x_4] \oplus DBS[x_8] \oplus C7k_{48} \oplus Bk_{49} \oplus 44k_{50} \oplus C7k_{52} \oplus Bk_{53} \oplus 44k_{54} \oplus C7k_{56} \oplus Bk_{57} \oplus 44k_{58} \oplus 9S[k_{60}] \oplus C9S[k_{61}] \oplus 49k_{62} \oplus 9k_{63} \oplus 49S[k_{63}] \oplus ES[k_{61} \oplus k_{57}] \oplus BS[k_{62} \oplus k_{58}] \oplus 9S[k_{60} \oplus k_{56}] \oplus ERC^3 \oplus C9RC^4 \oplus C9c_{22} \oplus 49c_{23} \oplus 9T_{62} \oplus DS[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62} ] \]

- \[ T_{47} = 52S[x_2] \oplus DAS[x_{10}] \oplus 49k_{48} \oplus 9k_{49} \oplus C9k_{50} \oplus 44k_{52} \oplus C7k_{54} \oplus Bk_{55} \oplus BS[k_{60}] \oplus 44S[k_{61}] \oplus C7S[k_{63}] \oplus S[k_{61} \oplus k_{57}] \oplus DS[k_{61} \oplus k_{53}] \oplus DS[k_{61} \oplus k_{57}] \oplus k_{10} \oplus 9S[k_{62} \oplus k_{58}] \oplus 9S[k_{62} \oplus k_{54}] \oplus 9S[k_{60} \oplus k_{54} \oplus k_{50}] \oplus BS[k_{60} \oplus k_{56}] \oplus BS[k_{60} \oplus k_{52}] \oplus BS[k_{60} \oplus k_{52} \oplus k_{56} \oplus k_{48}] \oplus DRC^1 \oplus DRC^2 \oplus DRC^3 \oplus 44RC^4 \oplus 44c_{20} \oplus C7c_{21} \oplus BT_{61} \oplus ES[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62} ] \]

- \[ T_{48} = k_{48} \oplus k_{52} \oplus S[k_{61}] \oplus S[k_{61} \oplus T_{0}] \oplus RC^4 \oplus RC^6 \]

- \[ T_{49} = k_{50} \oplus k_{54} \oplus S[k_{63}] \oplus S[k_{63} \oplus T_{3}] \]

- \[ T_{50} = k_{56} \oplus k_{60} \oplus S[k_{61} \oplus T_{0}] \oplus RC^6 \]

- \[ T_{51} = k_{58} \oplus k_{62} \oplus S[k_{63} \oplus T_{3}] \]

- \[ T_{52} = Ek_{52} \oplus Bk_{53} \oplus Dk_{54} \oplus 9k_{55} \oplus ES[k_{61} \oplus k_{57}] \oplus BS[k_{62} \oplus k_{58}] \oplus 9S[k_{60} \oplus k_{56}] \oplus ERC^3 \oplus DS[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62} ] \]

- \[ T_{53} = Dk_{52} \oplus 9k_{53} \oplus Ek_{54} \oplus Bk_{55} \oplus Dk_{60} \oplus 9k_{61} \oplus Ek_{62} \oplus Bk_{63} \]

- \[ T_{54} = Ek_{52} \oplus Bk_{53} \oplus Dk_{54} \oplus 9k_{55} \oplus Ek_{60} \oplus Bk_{61} \oplus Dk_{62} \oplus 9k_{63} \]

- \[ T_{55} = Dk_{52} \oplus 9k_{53} \oplus Ek_{54} \oplus Bk_{55} \oplus DS[k_{61} \oplus k_{57}] \oplus 9S[k_{62} \oplus k_{58}] \oplus BS[k_{60} \oplus k_{56}] \oplus DRC^3 \oplus ES[C9S[x_2] \oplus 44S[x_4] \oplus 44S[x_{10}] \oplus C9S[x_8] \oplus CBk_{56} \oplus 47k_{58} \oplus CBk_{60} \oplus 47k_{62} \oplus CBc_{20} \oplus 47c_{21} \oplus CBc_{22} \oplus 47c_{23} \oplus T_{61} \oplus T_{62} ] \]
C.2 Step-by-Step Procedure to Produce Equations for Substitution

The following details the step-by-step procedure to produce Equations (4.16) to (4.19). These equations are used in the initial substitution process to eliminate 8 state variables.
1. Describe $x_6$ in Eq. (4.1) in terms of $x_1, x_2, x_{10}$.

2. Describe $x_1$ in Eq. (4.2) in terms of $x_2, x_6, x_{10}$.

3. Substitute $x_6$ in Step 2 with $x_6$ in Step 1. Now $x_1$ is described only in terms of $x_2, x_{10}$ (Refer to Eq. (4.9)).

4. Substitute $x_1$ in Step 1 with $x_1$ in the previous step. Now $x_6$ is described only in terms of $x_2, x_{10}$ (Refer to Eq. (4.10)).

5. Describe $x_0$ in Eq. (4.3) in terms of $x_4, x_7, x_8$.

6. Describe $x_7$ in Eq. (4.4) in terms of $x_0, x_4, x_8$.

7. Substitute $x_0$ in Step 6 with $x_0$ in Step 5. Now $x_7$ is described only in terms of $x_4, x_8$ (Refer to Eq. (4.12)).

8. Substitute $x_7$ in Step 5 with $x_7$ the previous step. Now $x_0$ is described only in terms of $x_4, x_8$ (Refer to Eq. (4.11)).

9. Describe $x_3$ in Eq. (4.5) in terms of $x_2, x_4, x_5$.

10. Describe $x_{11}$ in Eq. (4.6) in terms of $x_8, x_9, x_{10}$.

11. Describe $x_9$ in Eq. (4.7) in terms of $x_8, x_{10}, x_{11}$.

12. Describe $x_5$ in Eq. (4.8) in terms of $x_2, x_3, x_4$.

13. Substitute $x_5$ in Step 9 with $x_5$ in Step 12. Now $x_3$ is described only in terms of $x_2, x_4$ (Refer to Eq. (4.13)).

14. Substitute $x_9$ in Step 10 with $x_9$ in Step 11. Now $x_{11}$ is described only in terms of $x_8, x_{10}$ (Refer to Eq. (4.14)).

15. Substitute $x_{11}$ in Step 11 with $x_{11}$ in Step 14. Now $x_9$ is described only in terms of $x_8, x_{10}$ (Refer to Eq. (4.15)).

16. Substitute $x_3$ in Step 12 with $x_3$ in Step 13. Now $x_5$ is described only in terms of $x_2, x_4$ (Refer to Eq. (4.16)).
Bibliography


[92] Pascal Junod and Serge Vaudenay. FOX: A New Family of Block Ciphers. In Helena Handschuh and M. Anwar Hasan, editors, Selected Areas in


[100] John Kelsey, Bruce Schneier, and David Wagner. Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In Neal Koblitz,


[115] Bon Wook Koo, Hwan Seok Jang, and Jung Hwan Song. On Constructing of a $32 \times 32$ Binary Matrix as a Diffusion Layer for a 256-Bit Block Cipher


[139] Jorge Nakahara, Jr., Pouyan Sepehrdad, Bingsheng Zhang, and Meiqin Wang. Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors,


[183] Lei Zhang, Wentao Zhang, and Wenling Wu. Cryptanalysis of Reduced-Round SMS4 Block Cipher. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, Information Security and Privacy, 13th Australasian Conference,


